Modeling the

Mechanical properties of Pinus Radiata

A Thesis

Submitted in partial fulfillment

Of the requirements for the Degree

Of

Master of Engineering (Mechanical)

In the

University of Canterbury

By

Samuel Tang Y.Y.

University of Canterbury
1998

Contents

Appendices **Pages**

Appendices **Pages**

Symbols

[ ]	Matrix notation
$[\ ]^T$	Transpose of a matrix
E	Elastic modulus
G	Shear modulus
σ	Stress
δ	Degree of freedom or displacement
γ	Shear strain
τ	Shear stress
ν	Possion's ratio
κ	Shear factor
H	Cellular height (mid wall to mid wall), tangential direction
W	Cellular width (mid wall to mid wall), radial direction
α	Cell off-set factor
D	Density of wood at conditioned RH and temperature
μ	Microfibril angle (MFA)
t	Thickness of cell wall
r	Area ratio
l	length of cell wall, radial direction
h	length of cell wall, tangential direction
b	depth of cell
U	Strain energy
V	Potential energy
χ	Total energy per unit volume
[N]	Shape function

[K]	Stiffness matrix
$[f]$	Force matrix
$[D]$	Stress- strain matrix
$[B]$	Shape matrix derivatives
M	Moment
I	Second moment of inertia

Note: Unless other wise stated within the text of the thesis, the notation of the symbol suffice.

Acknowledgment

The author would thank the guidance of Head of Department of Mechanical Engineering Prof. Jeremy Astley, his patience and understanding during the process of research and documentation of this work and the 'unforeseeable circumstances' during the course.

Special thanks to the group of researchers in the Mechanical Engineering Wood Group especially Jonathon Harrington and Kathrin Winkelman who so willingly contribute their time and wisdom to my limited understanding on Pinus Radiata. The author would also like to acknowledge the contributions of J. Neumann, Dr Robert Evans and K. Stol in part of the research material and the kind people at the computer laboratory who assisted in resolving application problems.

Lastly for my family in their understanding during this period of 'neglect' and the support given by them.

The research is funded by the Public Good Science Fund of New Zealand through research grant UOC 401.

Abstract

The study of a methodology to model the mechanical properties of Pinus Radiata takes place from the nanoscopic cell fiber [3] to the board scale level. Gibson and Ashy had generalized wood cells into hexagonal cells, previous work [2] extends the model specifically for Pinus Radiata. Unfortunately, earlier work was done on published data from various sources, not necessary related to Pinus Radiata nor from a common reference piece of characterized Pinus Radiata, making correlation with experimental work [4] difficult.

Further work was done by the author on a characterized sample of Pinus Radiata to correlate elastic properties with actual cellular geometry and experimental result. Critical geometrical parameters were studied for a feasible mathematical idealization as necessary parameters to further refine the FEM model.

Two approaches were used in evaluating board scale modeling; actual wood cells geometry and idealized hexagonal models. These models are extended to Growth ring model to predict growth ring mechanical properties and validated with experimental results as a preliminary Board scale model.

Stol had modeled Wood cells as hexagonal cellular material using Gibson's [1] analytical solution in his work with orthotropic wood wall's properties [2]. Stol's analytical solution neglects the longitudinal dimension, which in realism is closer to plate than beam.

Gibson and Asbhy's work on prediction of cellular properties analytically formed the basis of formulation analytical solution using energy method (Ritz's method) with plate type stiffness function and further extended with shear and longitudinal boundary coupling effect.

The plate analytical solution was validated by FEM to be in close agreement, within a 5% error. The model based on real cell geometry and its equivalent regular array of

identical cell has broad agreements with experimental values. Further refinements of this model are important steps in the development of a definitive model.

Preliminary work on a growth ring FEM model is important as part of the preparation for a board scale model, however further refinements would be necessary for definitive board scale FEM model.

Chapter 1.

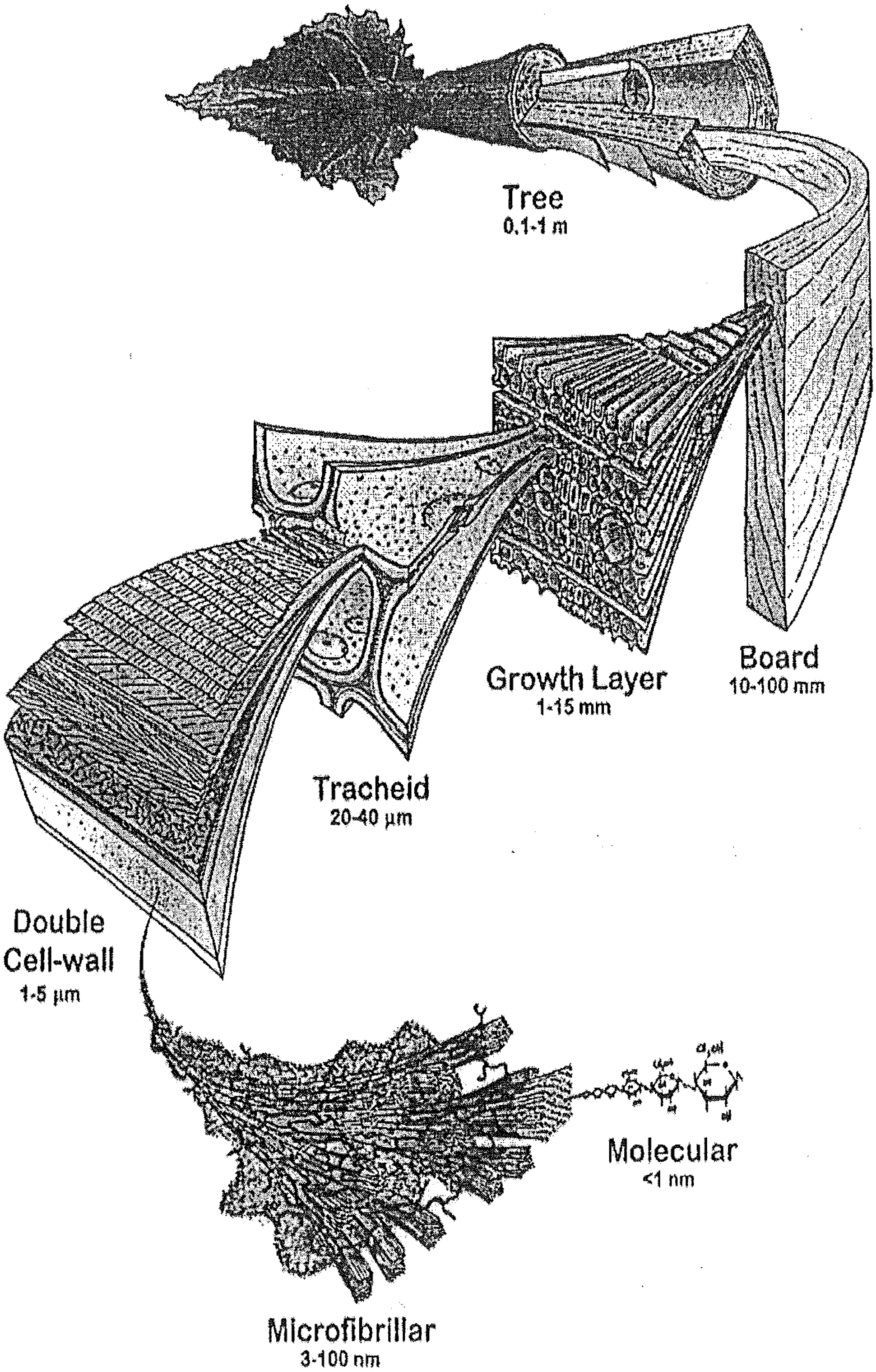
Introduction to Mechanical Engineering Wood Technology (MEWG) Group's work

The Wood Technology Research Center at the University of Canterbury was fully established at the end of 1996 as a means of information exchange among staff engaged in wood-related research, to facilitate the shared use of research facilities and enhance research programs and to promote technology transfer to end users. Staff from three Engineering Departments, Chemistry, Forestry, Plant and Microbial Sciences and Zoology participated in the Center's research activities. The Center organises seminars and has links with the New Zealand Forest Research Institute and collaboration with major research centers around the world on research projects.

The MEWG was formed as part of the Wood Technology Research Group focused mainly on work related to mechanical properties of Pinus Radiata. The MEWG is housed within the Mechanical Engineering Department of the University of Canterbury. The work done by MEWG is in three main areas, microscopic research [5], mechanical elastic property modeling [2] and micro-specimen measurement techniques [4]. All three areas are worked in parallel by separate researchers.

1.1 Wood Cell Geometry

In general, a tree consists of a single wooden stem covered with a layer of bark. The wood material in the stem consists of wood cells. The shape and size of the cells depends upon their physiological role in the tree. The mechanical performance of Softwood are found to be highly dependent on the nested sequence of the microstructural cells which occurs at scales ranging from submicroscopic to board scale. These are illustrated in figure 1.1.



[1] Illustrated by J.J. Harrington, member of MEWG

The largest proportion of the stem volume consists of wood cells called tracheids, fibers or grain, which are long thread-like cells with their longitudinal axes parallel to the central axis of the tree, the radial axes aligned radially from pith to bark and the tangential axes being tangential to the radial and longitudinal axes. These axes are the primary axes when cells are being studied in isolation. The main functions of these cells in the stem are mechanical support and fluid transport. Another type of cells, called parenchyma cells, are also found in the tree. These are the basic elements in the so-called raywood rays, which are bands of cells oriented perpendicular to the longitudinal fibers and extent in radial direction. Their function is mainly that of nutrient storage and insignificant from a mechanical performance viewpoint.

Since the tracheids gives the wood material strength and stiffness, the mechanical properties and their orientations in sawn timber are of great importance. Cell production in a tree takes place just inside the bark, where a ring of so-called cambial cells produces the tree's wood cells on the inside and its bark cells on the outside. The fibers form a cylindrical layer called growth ring. A growth ring is normally produced during one year and consists of two distinct parts called early wood and late wood. The late-wood being denser and darker than the early wood makes the growth rings visible to the naked eye. Typical early-wood cell wall is 2-4 μm and late wood; 3-7 μm with density variation in the range of 300-800 kg/m^3.

The cell can be divided into three parts; the cell wall, the cell lumen and the middle lamella. The cell wall is the structural part of the cell and the cell lumen is the cavity of the cell where the fluid transport occurs. The middle lamella is a surrounding medium around the cell wall, which interconnects the cells. For further information on wood cells and their internal structure, see Bodig and Jayne [14], Kollman and Cote [15], Dinwoodie [16] and Kinnimonth [17].

The behaviour of wood is strongly affected by the environmental conditions. During changes in moisture content, considerable shrinkage or swelling of the material occurs. The moisture related behaviour of wood is a highly researched area (See researches related to moisture e.g. [19-23]) and for the purpose of simplicity and structural application, all wood properties are assumed and conducted at 12 % moisture content. In general, optimal mechanical strength is found at 12% moisture content and used commercially as a standard for drying of lumber.

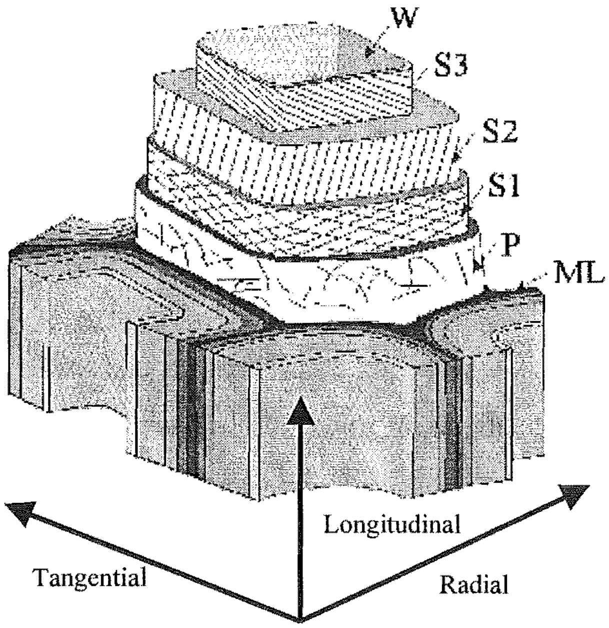


Figure 1.2. Single Wood tracheid

Wood cells (figure 1.2) are honey-comb type structures with complex composite double cell wall in varying proportion of cellulose, hemi-cellulose and lignin within a microfibril unit. Each double cell wall pair has distinct orientation of microfibrils (microfibrils angle or MFA) and is represented by a laminate of seven orthotropic plies. Each ply S1, S2, S3 and the middle lamella (ML and P or CML) has distinct chemical composition and mechanical properties and susceptibility to moisture content.

1.2 Characterisation of Orthotropic Elastic Material[2]

The stress within an orthotropic elastic material is defined by six stress components. In a curvilinear co-ordinate system with local orthogonal axes 1,2 and 3, these are direct stress σ_i (i= 1..3) which represent normal force intensities acting on each face of an elemental cube and the shear stresses τ_{ij} (i,j=1..3) which represent tangential force intensities on the same planes. The corresponding strain components consist of direct strains e_i and shear strains γ_{ij} (i,j =1..3). For a linearly elastic material, the relationship between stress and strain is given by:

$$\sigma = D\,[\,e - e_{mc}] \text{ or } e = C\sigma + e_{mc}, \text{ where } C = D^{-1} \qquad (1.1)$$

Here σ and e are column vectors containing the six independent components of stress and strain respectively, and e_{mc} is a vector of initial strains which in this case arise from shrinkage and expansion due to a variation of moisture content, Δmc. The matrices D (stiffness matrix) and C (compliance matrix) are symmetrical and contain 21 independent terms. In the case of an orthotropic material, symmetry planes exist perpendicular to the orthogonal axes and the number of independent terms then reduces to 9. Substituting the three orthogonal directions for wood, namely longitudinal(l), tangential (t) and radial (r) into the relationship 1.1 yields:

$$\begin{bmatrix} \varepsilon_r \\ \varepsilon_t \\ \varepsilon_l \\ \gamma_{rt} \\ \gamma_{rl} \\ \gamma_{tl} \end{bmatrix} = \begin{bmatrix} \frac{1}{E_r} & -\frac{\nu_{lr}}{E_t} & -\frac{\nu_{lr}}{E_l} & 0 & 0 & 0 \\ -\frac{\nu_{rt}}{E_r} & \frac{1}{E_t} & -\frac{\nu_{lt}}{E_l} & 0 & 0 & 0 \\ -\frac{\nu_{rl}}{E_r} & -\frac{\nu_{tl}}{E_t} & \frac{1}{E_t} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{G_{tl}} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{G_{rl}} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{G_{rt}} \end{bmatrix} \begin{bmatrix} \sigma_r \\ \sigma_t \\ \sigma_l \\ \tau_{rt} \\ \tau_{rl} \\ \tau_{tl} \end{bmatrix} + \begin{bmatrix} \alpha_1 \Delta mc \\ \alpha_2 \Delta mc \\ \alpha_3 \Delta mc \\ 0 \\ 0 \\ 0 \end{bmatrix} \qquad (1.2)$$

The Poisson's ratios ν_{ij} and ν_{ji} are not independent being related by the symmetry condition

$$\nu_{ij} / E_i = \nu_{ji} / E_j \qquad (1.3)$$

[2] Article from IUFRO conference 'Microfibril angle in Wood' at Westport, NZ 1998 [28]

1.3 Modeling Techniques[3]

To analyse the cell wall properties, then a single cell and extend it to model its macroscopic performance is impractical and a sequence of homgenisation procedures is necessary. Each of these involves the formulation of an "equivalent" homogeneous material whose mechanical characteristics are derived from a more detailed heterogeneous model. This yields a nested sequence of smaller and more tractable simulation exercises. In the case of softwood, the progression from molecular to growth ring scales is achieved in three steps. These are:

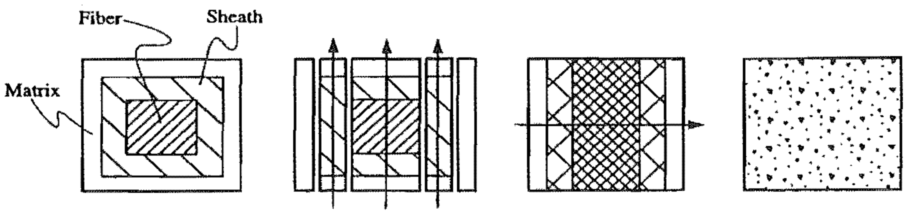
a. Homogenization of a heterogeneous microfibril unit to give equivalent homogeneous material for an oriented element (MFA) of the cell wall lamellae.

b. Homogenisation of a heterogeneous distribution of oriented lamellae to give a homogeneous material for a cell-wall layer.

c. Homgenisation of a heterogeneous arrays of cells to give a homogeneous material for early .

wood and late wood or an entire growth ring.

A homogenisation technique (Representative volume element -RVE) is used to model a microfibril structural element (figure 1.3). The central cellulose core forms the fiber phase of the composite. Surrounding the core is a sheath of cellulose and polyose chains aligned in the microfibril direction, the remainder of the RVE is occupied by an amorphous matrix. This is itself a mixture of lignin and polyose. A second RVE (not shown) is used to provide its properties.



(i) Nanostructural RVE (ii) 1st Pass (iii) 2nd Pass (iv) Equivalent Material

[3] Article from IUFRO conference 'Microfibril angle in Wood' at Westport, NZ 1998 [28]

Figure 1.3 Multipass homogenisation of microfibril RVE [3]

	Fibre	Sheath	Matrix
E_l	*128*	*16*	*4.2*
$E_t(=E_r)$	*18*	*6.8*	*4.2*
ν_{tl}	*.05*	*.1*	*.31*
$G_{lt}(=G_{lr})$	*6.0*	*2.38*	*1.6*
G_{rt}	*6.0*	*2.38*	*1.6*

Table 1.1 Constituent Elastic Properties (GPa) [28]

Different RVEs of this general type are required to model different layers within the cell walls and the techniques used is developed by Chou and Carleone [24] and discussed in earlier work [2] by the MEWG. These results are in close agreement with models done by Persson [26] and Koponen [27].

	E_t	E_r	E_l	ν_{tr}	ν_{lt}	ν_{lr}	G_{rl}	G_{lt}	G_{rt}
S3	*8.43*	*7.98*	*50.36*	*0.39*	*0.33*	*0.32*	*2.65*	*3.00*	*2.68*
S2	*9.85*	*9.16*	*63.96*	*0.39*	*0.33*	*0.33*	*3.02*	*3.38*	*2.96*
S1	*8.54*	*8.02*	*53.10*	*0.38*	*0.33*	*0.32*	*2.66*	*3.02*	*2.66*
CML	*5.07*	*5.12*	*18.43*	*0.38*	*0.31*	*0.31*	*1.78*	*2.11*	*1.88*

Table 1.2 Computed elastic constants for microfibrillar RVEs [28] (12% mc, moduli in GPa)

The variation of microfibrillar orientation within each cell-wall layer is modeled by a second homogenisation in which a prescribed probability distribution $f(\gamma)$ ($-\pi/2 \leq \gamma \leq \pi/2$) is assumed for the microfibril orientation γ about a mean microfibril angle μ (MFA). Within S1 and S3 layers, the microfibrils are assumed to be distributed about a mean value of 70° with a standard deviation of 12.5°.

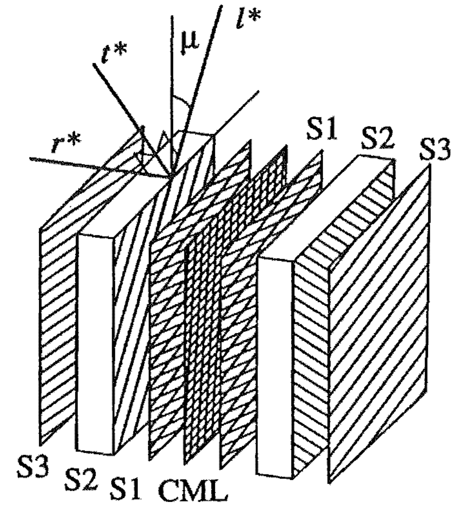


Figure 1.4 Homogenisation of cell wall.

In the S2 layer, a normal distribution is assumed with a standard deviation equal to one third of the mean MFA. This closely followed Cave's model [18, 19, 20]. The CML is assumed to be uniform so that homogenisation gives an isotropic material in the l^*-t^* plane.

	E_{t^*}	E_{r^*}	E_{l^*}	$\nu_{t^*r^*}$	$\nu_{l^*t^*}$	$\nu_{l^*r^*}$	$G_{l^*r^*}$	$G_{l^*t^*}$	$G_{t^*r^*}$
S3	*8.43*	*8.07*	*45.4*	*0.40*	*0.44*	*0.27*	*2.65*	*4.67*	*2.68*
S2(μ=0°)	*9.85*	*9.16*	*63.96*	*0.39*	*0.33*	*0.33*	*3.02*	*3.38*	*2.96*
S2(μ=10°)	*9.83*	*9.16*	*63.47*	*0.39*	*0.35*	*0.32*	*3.02*	*3.57*	*2.96*
S2(μ=20°)	*9.76*	*9.18*	*62.01*	*0.39*	*0.40*	*0.30*	*3.02*	*4.09*	*2.96*
S2(μ=30°)	*9.78*	*9.23*	*59.69*	*0.38*	*0.47*	*0.27*	*3.01*	*4.87*	*2.96*
S2(μ=40°)	*9.88*	*9.28*	*56.68*	*0.37*	*0.54*	*0.23*	*3.01*	*5.80*	*2.97*
S1	*8.55*	*8.10*	*47.80*	*0.37*	*0.50*	*0.24*	*2.66*	*4.80*	*2.67*
CML	*9.55*	*5.44*	*9.55*	*0.30*	*0.30*	*0.30*	*1.83*	*3.67*	*1.83*

Table 1.3 Computed elastic constants for homogenised cell wall layers (12% mc)

The values in Table 1.3 are the cell-wall properties that are essential for finite element modeling of the trachied cells to give macroscopic elastic properties. The commercial FE code ANSYS is used to construct and analyse such models. Each double cell wall by volume is occupied by 13% S1, 60% S2, 9% S3 and 19% CML. The MFA for S1 and S3 were assumed as 70° and S2, being the structural layer was assumed as a variable parameter μ (MFA) as a primary input to the macroscopic FE model. The experimental determination of S2 MFA is carried out as a separate research project by the MEWG [4].

1.4 Preliminary work

Earlier work by Stol and others [2] in mechanical elastic modeling had done a remarkable job analysing the complex geometry of wood cells of Pinus Radiata by extracting geometrical information from micrographs and constructing representative finite element models.

Two different methodologies were considered; one method assumed wood to be a regular structure made of repeated units, and the other which assumed an irregular array of cells reconstructed by digitised micrographs. Both models make use of statistical information such as mean cell width, mean cell height, area ratio and mean density found from digitisation of micrographs or published literatures. The resulting elastic constants were compared to typical wood properties published for Stika Spruce and Norway Spruce; both similar softwood species due to the absence of published data for Pinus Radiata.

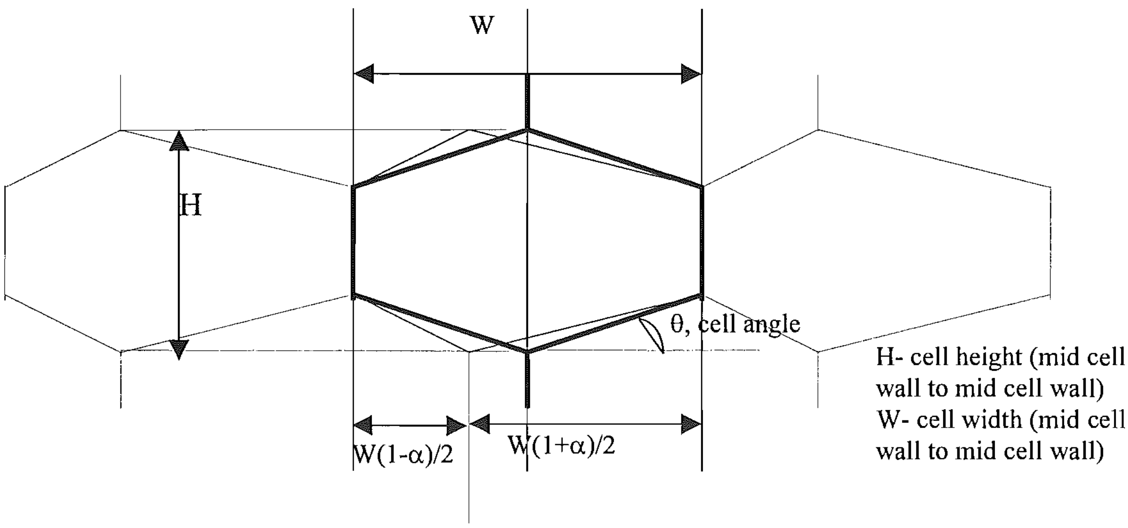


Figure 1.4 Regular model with offset factor α

Regular modeling starts with analysis of numerous micrographs of early wood cells using image analysis software such as METAMORPH to evaluate the lumen height, width and area to determine geometrical parameters such as theta, area ratio r and offset factor α. A detailed mathematical relationship of various parameters is enclosed in the appendix.

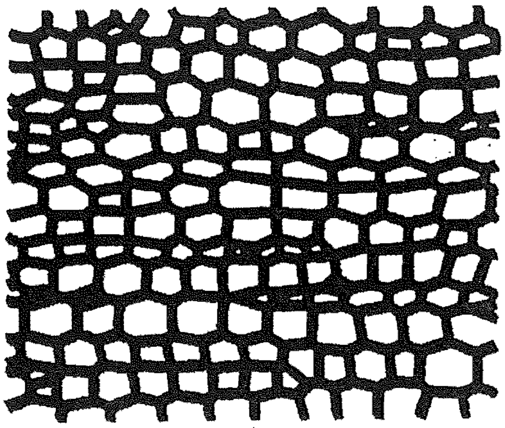


Figure 1.5 Micrograph derived irregular or skeletal model

Irregular model or skeletal model is constructed by plotting lines using the CML as the boundaries of polygons from actual micrographs, the skeletal structure is the FE model. In theory, this would represent the closest to the physical structure and the regular model would be an ideal as a mathematical model, given the severe limitations of generating skeletal model. Both models were constructed with ANSYS SHELL91, thick-100 layered 8-noded elements which include the effects of in-plane membrane strains and shear deformation.

By comparing results obtained from both models, the regular model is found to closely approximate the skeletal model for cell angle θ equal to 15.9 degrees and an offset α equal to 0.65. This is in agreement with published moduli ratio of 1.7 [29]. The two values for θ and α will be utililised later in the growth ring model.

With the advancement of the research work by the other two groups, namely nanoscopic modelling and experimental moduli measurements, and as part of the research to refine the mechanical model, it is imperative that the three groups need a common reference piece of Pinus Radiata as mathematical modelling was done independent of both the experimental researches.

To further the work, it is necessary to use a common reference to correlate the mathematical regular cellular model with experimental results using a piece of Pinus Radiata sample grown in Southland of New Zealand. This work forms the scope of this research thesis which are in two parts:

a. To duplicate Stol's single cell FEM model using the Silviscan and micrographic geometrical data to correlate with experimental results in the following steps:
    1. Growth rings geometrical data from Silviscan are analysed and variations idealised for a typical growth ring.
    2. Thin section samples are analysed to validate the Silviscan data.
    3. Develop a growth ring FEM model using the above results and correlate with small sample experimental measurements.

b. Formulate analytical solution using Energy method for hexagonal cellular models:
    1. Validate Gibson and Ashby's simple and advanced model using Energy method and beam element.
    2. Extend the beam analytical solution using plate elements with bending and membrane force considerations.
    3. Extend the plate analytical solution to include bending, membrane force and shear effect considerations.
    4. Extend the plate analytical solution to include bending, membrane force, shear effect and longitudinal strain effect considerations.

Chapter 2

Characterisation of Reference Pinus Radiata specimen

Validation of the models for wood mechanics being developed requires structural and compositional data for wood whose elastic properties are also known. The wood to be used for model verification will come from within the trunk of a 25 year old Pinus radiata tree grown in Southland, New Zealand. Since all the different tests should be to be conducted on the same piece of wood, or at least on pieces from a restricted locale, all the steps from the first cutting to the last test need to be planned thoroughly [32]. Cell shape, size and length, cell-wall thickness, and microfibrillar orientation constitute the structural features for which quantitative data is required if the cellular mechanics models are to be validated.

There are different methods of obtaining cell geometrical information. The geometrical information can be obtain either directly through thin section microscopy or through a recently developed X-ray Silviscan procedure [6]. Both methods were used during the research by MEWG and the author and the merits of each methods are being evaluated at the time of writing. However, both methods have close agreement as far as geometrical results are concerned.

2.1 Direct measurement method

In the earlier method, cell shape, size and length, and cell-wall thickness can be determined using a light microscope in conjunction with a digital image analyser. Thin transverse sections (15μm) from resin embedded samples give high resolution images and reveal the original cell shape, size and cell wall thickness. The cell length can be determined from macerated cells. Thin microtomed sections allow the preparation of samples with smooth surfaces and their cells oriented in the main directions (L, T or R), as is required for the mechanical testing. Shavings (5μm thick) are kept for structural study. Cutting is also done with a guillotine using a razor blade. The current method allows cuts of up to 3mm in depth to be made. The wood is soaked in water before cutting. After being cut it is treated carefully with acetone and ethanol to remove water and extractives, such as resin. Common histological methods, such as staining, resin embedding and maceration, are then used to reveal the structure. This process takes up to two weeks. The prepared sample is then subjected to back-light microscopy and digitised and analysed with image analyser. A typical image is showed in figure 2.

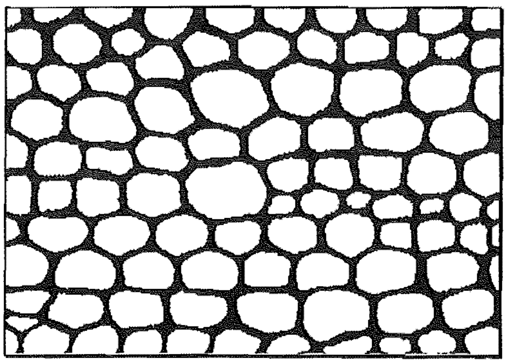


figure 2. Successful image of cell sections

The microfibrillar angle were analysed by Winkelman by the iodine-method [31] or Stained-cell method [1]. The method is time consuming and requires careful specimen preparation, similar to the method for cell geometry measurement. Cuts along the T direction are taken from the prepared specimens to slice through the plies of the cell-walls. Iodine stained microfibrils are manually measured under high magnification. Stained-cell method gives a more direct measure of microfibril angle, and permits very localised observation for individual cells.

Direct measurement method is laborious and time-consuming. The process is highly dependent on the quality of the wood specimen and preparation method. Good high contrast micrographs acceptable for image processing are very difficult to prepare. The quality of the wood cell structure is equally important, there should be no rays, large pits, badly deformed or unbounded cells. These imperfections are rejected by image processing software and often meaningful results are difficult to obtain. The shortcoming is further aggravated when studying latewood because of the scarcity of latewood. Early attempts by the author to characterise the complete growth ring variations were abandon because of the difficulties involved in obtaining perfect high contrast micrographs and the large quantity of samples required for 21 growth rings. However sufficient micrographs were prepared for growth ring 14 to validate the accuracy of Silviscan data.

MFA measurements are equally difficult because the readings are highly influence by local defects such as pits, rays, knots and local imperfections. These imperfections cause localised distortion of the microfibrils and the sites chosen for MFA measurement may not be representative of the average MFA of the cell or growth ring. Choice of measurement sites are very important and at microscopic level the choice of sites are limited.

2.2 Silviscan Method[4]

Silviscan [6] was developed at Commonwealth Scientific and Industrial Research Organisation, Austrialia, (CSIRO) Forestry and Forest Products for the rapid measurement of Pinus Radiata microstructure. The system make use of a co-ordinated x-ray micro-densitometer, diffractometry and image analyser to rapidly scan across a wedge of sample wood made across growth rings of a tree.

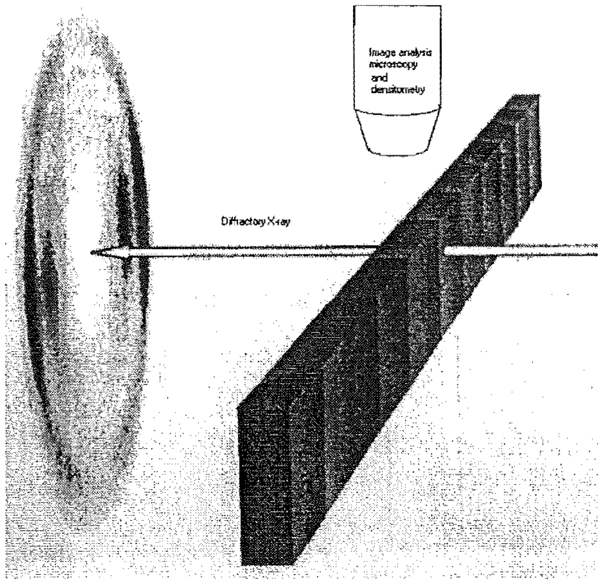


Figure 2.1 X-ray Silviscan

The sample wedges are dried and conditioned for at least 18 hours at 23°C and at 50% relative humidity. Tracheid diameter profiles and the orientation of rings and rays with respect to the sample were first determined by image analysis [42]. The x-ray densitometer then tracked the ring orientation profile while measuring the density. Mathematical combination of the density profiles with the tracheid diameter profiles

[4] Performed by Dr Robert Evans, CSIRO, Division of Forest Product, Victoria, Australia

gave coarseness, wall thickness and specific surface area profiles. The relationships for tracheids of rectangular cross-section are

$$P=2(R+T) \quad (2.1)$$

$$C=RTD \quad (2.2)$$

$$S=(2P-8W)/C \text{ including the lumen} \quad (2.3)$$

$$W=P/8-1/2\sqrt{(P^2/16-C/d)} \quad (2.4)$$

where W= tracheid wall thickness

P= tracheid external perimeter

R= radial tracheid dimension (pith to bark direction)

T= tangential tracheid dimension (parallel to the rings)

D= wood density at conditioned relative humidity (RH) and temperature

d= tracheid wall density (1500 kg/m^3)

C= tracheid coarseness (mass per unit length)

The measurement of MFA angle is performed by another x-ray diffractometry [41] scan measuring the microfibril axis in the b-direction. Silviscan-2 uses a wide-angle x-ray area detector to acquire 2-dimensional diffraction pattern without sample rotation. The detector is able to identify the 002 diffracted energy to correlate with the MFA. X-ray diffractometry (XRD) has been correlated relatively well with polarised light microscopy [44] and pit angle measurement [43], both direct measurement techniques.

The XRD method assumed that the variance (S^2) of the (002) azimuthal diffraction profile is related to the MFA(μ) and the variance (σ^2) of he microfibril orientation distribution:

$$S^2 \approx \frac{\mu^2}{2} + \sigma^2 \quad (2.5)$$

This equation is valid for regular cross-section from triangular to circular. Meylan [45] showed that the diffractional pattern is unaffected by spatial position within the path of the beam and Cave [46] found the correlation of XRD estimates could be reconciled with microscopy MFA if the standard deviation in micorfibril orientation is assumed to be proportional to the mean.

$$S^2 \approx \frac{\mu^2}{2} + \sigma_{mult}^2 + \sigma_{add}^2 \qquad (2.6)$$

Using a proportionality factor k (coefficient of variation), $\sigma_{mult} = k\mu$ giving

$$S^2 - \sigma_{add}^2 \approx \mu^2\left(\frac{1}{2} + k^2\right) \qquad (2.7)$$

If k=1/3 as used by Cave [46], then equation (2.7) gives

$$\mu \approx \sqrt{\frac{18}{11}\left(S^2 - \sigma_{add}^2\right)} \quad \text{or} \quad \mu \approx \sqrt{\frac{18}{11}} S_{corr} \approx 1.28 S_{corr} \qquad (2.8)$$

where S_{corr} is the measured standard deviation of the diffraction profile corrected for additive variance (after baseline and absorption correction).

2.3 General cell geometry

Clear wood from the stem of a softwood tree is composed most part of axially aligned tracheid cells. These have an aspect ratio (length to width) of the order of 100:1. They are created in the vascular cambium of the tree in a radial expansion to form irregular honeycomb of prismatic tubes, which possesses a distinct tangential diameter, at least over a growth ring. The tangential diameter seems to be independent of the growth rate within a growth ring whereas the radial diameter is dependent on the growth rate at the particular time of the year. The tracheid cells of the Pinus radiata sample under study are typically 20-60μm and length of 2-4mm depending on the location within the tree [37].

The biological definition of a growth ring in softwood is the position in where the growth process converts from late-wood with thick fiber walls and a small lumen diameter ratio to early wood with thin walls and a large lumen diameter ratio. As found from a microscopic investigation of a sample, this transition takes place in a very distinct and narrow region. In the rapidly growing early wood, cells have relatively large radial diameters and thin walls, in latewood, the radial diameters decrease and cell walls are thicker. The change in morphology makes growth ring visible. Typically early wood wall thickness is 2-4 μm and that of late wood 3-7 μm.

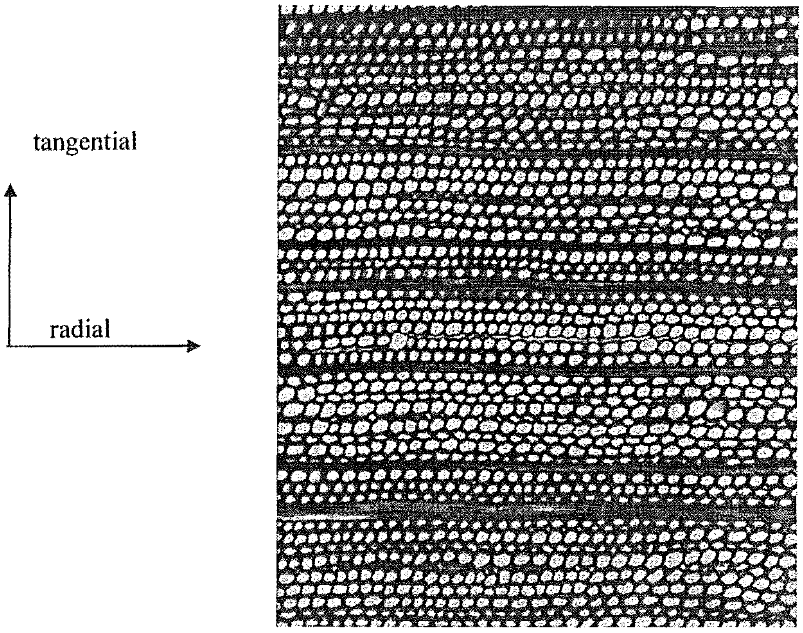


Fig 2.2 Typical growth ring micrograph of early wood

In microdensitometry this transition is easily seen as a rapid change in density across the growth ring and varies from 300 kg/m^3 in the early wood to 800 kg/m^3 in the latewood. The ratio of local density to the density of cell wall material is called area ratio because it is equal to the ratio of cell wall area to ratio of the total cell area. This parameter is of prime importance because it defines the structural cross section of the wood cells.

The raw data from X-ray Silviscan is shown in Table 2.1. The 'position' relates to the distance measured from the start of the specimen and is correlated to the growth ring number. To start analysing the data, we need to segregate the data into 'growth rings'. With the knowledge that density variation is very distinct when transiting from early to late wood, we can identify the start of the growth ring, for example Growth Ring 14 when the density starts declining (transition from late wood of growth ring 13 to early wood of growth ring 14) and the end of Growth ring 14 when the density climaxed. This block of information is then labeled as 'Growth ring 14'.

X-ray Silviscan data

Position (kg/m3)	Density (kg/m3)	Rad diam (um)		Coarse (ug/m)	Wall thick (um)	Spec surf (m2/kg)	
140	763.5	27.6	24.8	523.3	3.91	200.3	
140.05	775.6	26.4	24.8	503.3	3.86	203.5	
140.1	781.3	24.7	24.9	474.1	3.75	209.5	* Start of growth ring 14 denoted by the peak density of ring 13's latewood.
140.15	741.9	23.5	25	457.2	3.72	211.9	
140.2	633.3	29.4	25.5	454.4	3.16	241.9	
140.25	515.1	35.9	26.1	471.3	2.79	262.8	
140.3	460.6	39.5	26.7	481.7	2.64	275	
....	....	....	....	....	....	.....	
...	....	....	....	....	....	.....	
156	729.3	26.3	25.1	476.8	3.6	215.6	
156.05	744.6	24.2	25	445.1	3.52	221.2	* End of growth ring 14 denoted by the max. density.
156.1	738.6	22	25	413.9	3.44	227	
156.15	663.1	23.2	25.4	404.3	3.2	240.4	
156.2	547.8	30.2	26.2	415.2	2.74	271.3	

Table 2.1 Typical raw data from Silviscan

If we plot the density against the distance, the typical plot would be very similar to observed density variation for a growth ring [42, 8].

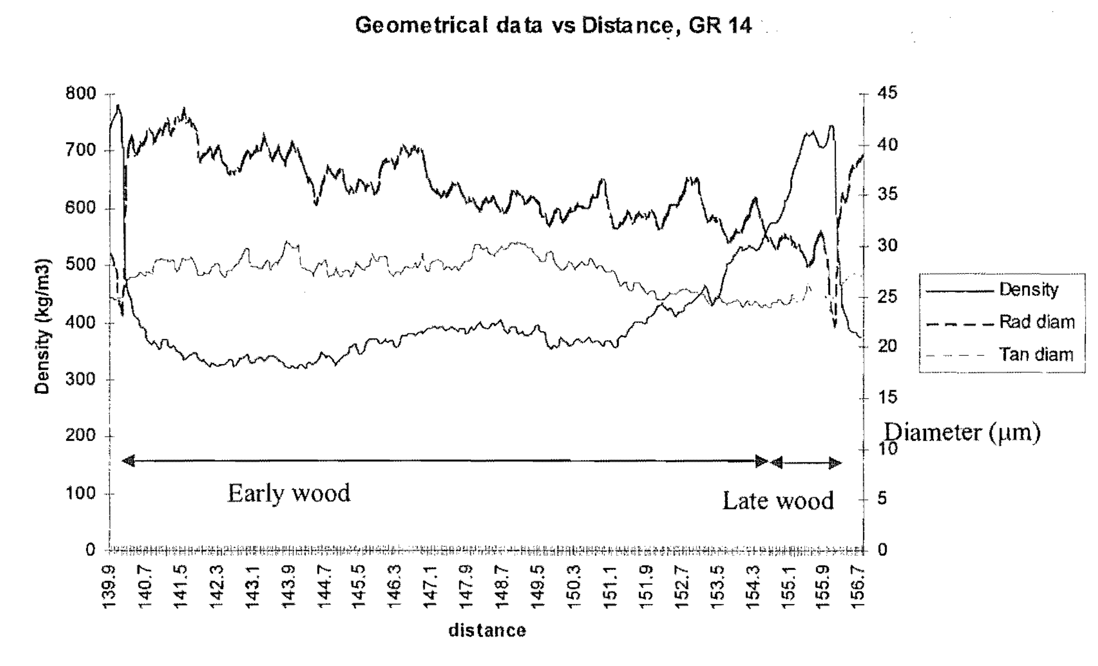
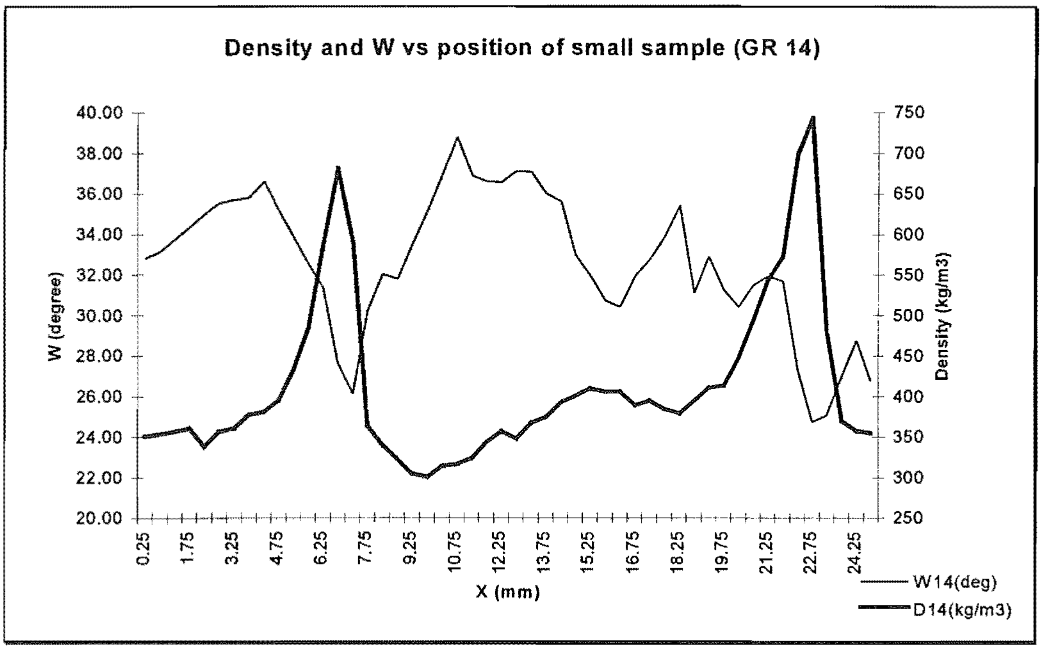


Figure 2.3 Growth ring 14 Silviscan

The MFA (W) values are obtained in a separated X-ray diffraction scan and again correlated by distance. The result for growth ring 14 is showed below, correlated with the density on a coarser graduation that the figure above.



W14-MFA values via XRD

D14-Microdensitory metry values

Figure 2.4 MFA variation in GR 14

The region of particular interest is the transition from the early wood to late wood. If the data are idealised for each growth ring, the transition would not be readily visible. To resolve this, the growth ring distance is recalculated using the start of GR14 as zero, then normalised to the growth ring width and offset by a normalised distance of 0.15. This translation will enable the full early-wood to late-wood and back to early-wood be clearly visible for idealisation. Various other geometrical data are also normalised to the respective average value over the growth ring.

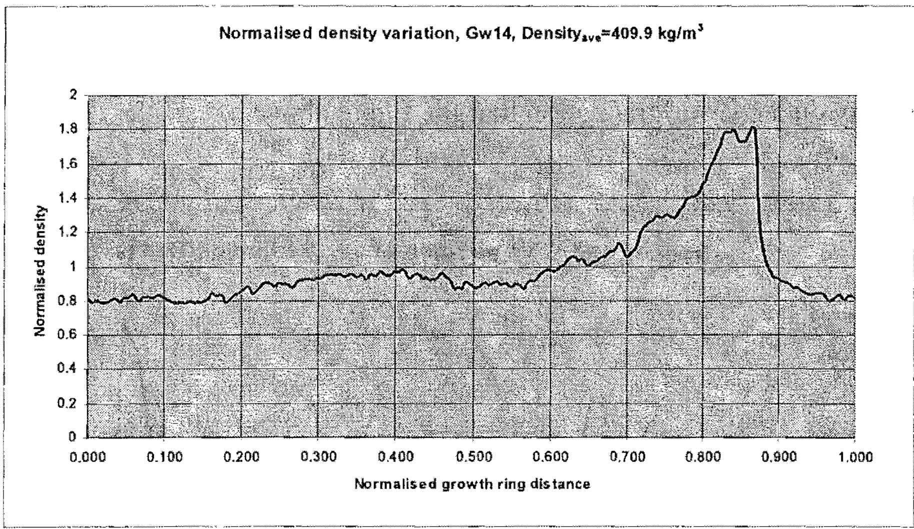


Figure 2.5 Offset normalised density variation of growth ring 14

Norm. dist	density	Rad. dia	Tang. Dia	MFA
0.000	0.822366	1.075868	1.022313	1.019277
0.003	0.806246	1.095234	1.012409	1.017387
0.007	0.789812	1.110937	1.007641	1.013878
0.010	0.7985	1.130041	1.012776	1.010369
0.013	0.806538	1.125592	1.009842	1.024753
0.016	0.812757	1.104394	1.016444	1.039141
0.020	0.809142	1.102039	1.026715	1.050474
0.023	0.800678	1.114601	1.021947	1.061811
0.026	0.813498	1.114862	1.032217	1.053602
...	...	...	...	...
0.980	0.813183	1.166418	1.074034	0.93453
0.984	0.790395	1.167465	1.05496	0.920647
0.987	0.80436	1.160922	1.086139	0.929478
0.990	0.780494	1.199131	1.08834	0.938309
0.993	0.759098	1.192327	1.049091	0.928728
0.997	0.769471	1.175578	1.048357	0.919152
1.000	0.785411	1.170344	1.049824	0.941745
average	445.4099	38.21121	27.26177	23.33888

Table 2.2 Growth ring 14 normalised geometrical data

This example shows density for growth ring 14 and the rest of the geometrical data follows the same typical method of analysis and presentation, a sample data is shown in Table 2.2.

2.4 Silviscan of Reference Wood Sample

The reference wood sample was prepared from a 25 years old Pinus Radiata grown in Southland and sent to CSIRO for X-ray Silviscan. The data from Growth ring 4 to 24 were analyzed for trends as inputs to the numerical FE model.

2.4.1 Analysis of data

For comparison of various cell properties, each growth rings are isolated beginning from the start of the early-wood to the end of the late-wood and offset as described in the earlier section. The measured quantities are normalized and plotted in a non-dimensional form. All 21 growth ring's geometrical data points are superimposed on each other to establish the variations and trends within the growth rings. Once the trends are established, approximations to the trends are made to define a piecewise linear approximation.

It is necessary to note that cell diameters are measured from the middle wall thickness of one cell to the middle wall thickness of the corresponding cell and should not be confused with lumen diameters associated with high contrast imagery processing techniques. These are obtained by X-ray absorption variations when the beam transverse across a cell.

Microfibril angle (MFA) are measured by 2 techniques, Silviscan W values which are assumed as an indication of MFA (averaged over the x-rayed section) and the Stain-Cell values which are direct and localized measurements of MFA.

Some 41,000 data points are analyzed in the process for growth ring 4-24, a total of 21 growth rings.

2.4.2 Trend of density across growth ring

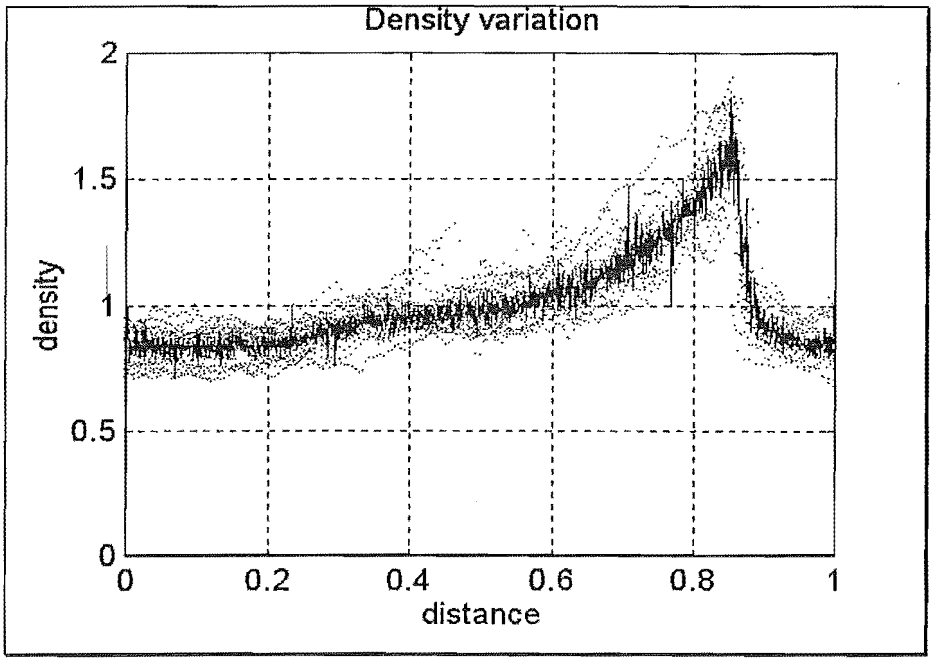


Figure 2.6. Data points and averaged density

Figure 2.6 show the superimposed growth ring 4-24 normalised density points as dots on the graph and the solid line being the average of corresponding points at each normalised distance within each growth ring. The average acts a filter to removed short-term variability of the data points due to other external effects such environmental effect and compression wood that may be present at one particular growth ring.

The average plot of normalised density versus the normalised distance shows a very consistent density trend that would be expected in growth rings of Pinus Radiata. The early wood begins with the lowest density of 350-400 kg/m^3 and gradually increases to the late wood of 650-700 kg/m^3. Early wood have large cells and thin walls are expected to have low area ratio which corresponds to low density and the late-wood with its thick wall; the highest density. The gradual increase in density indicates the growth rate of Pinus Radiata may be temperature dependent with very slow growth in late wood (winter). The onset of spring re-activates the growth mechanism; the growth rate increases drastically and early wood starts forming. The absence of any large humps between early wood and the peak of the late wood (distance 0.2 to 0.6) indicates the wood is relatively free from compression wood, except for certain growth rings. A typical density profile affected by compressed wood such as growth ring 8 is shown in Figure 2.7. Compression wood is formed when a tree subjected to unusual stress leads to plastic deformation (compression in most case) within the tracheids and unusually high density is found between early and late wood. The 'compressed wood' normally have a smaller radial diameter and hence higher density.

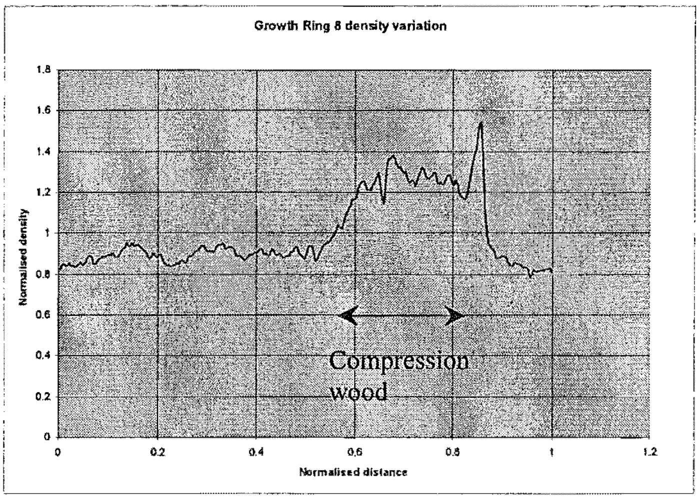


Figure 2.7 'Compression' wood in Growth ring 8, identified by the abnormally high density.

All 21 growth rings exhibit the same trend and the maximum density occurs at a distance of 0.85 of the growth ring. This is the point at which all other cell geometry parameters climaxed as well, except for tangential diameter being relatively constant.

An approximation is made to represent density as a function of the growth ring distance (see figure 2.9). The corner co-ordinates necessary to reproduce the approximation are also shown in the figure

Variations of average density of each growth rings is plotted in figure 2.8. The average density shows that density remains relatively constant over time and if any increase at a rate of 2.5 kg/m^3 per year with variation between +-18% from the mean of 449.87 kg/m^3. This variation may be considered in boardscale modelling.

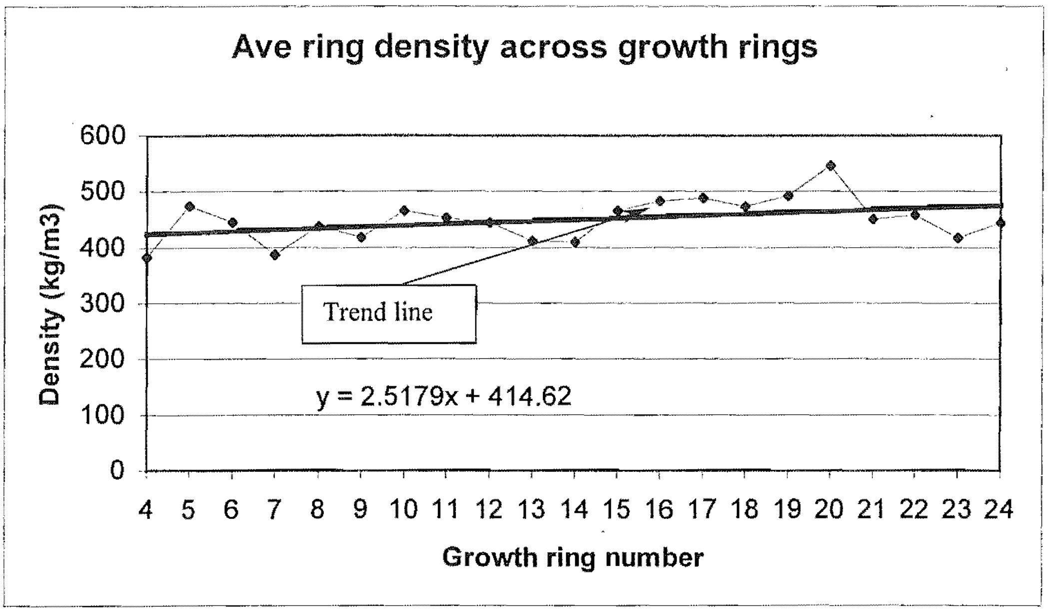


Figure 2.8 Density variation across growth rings

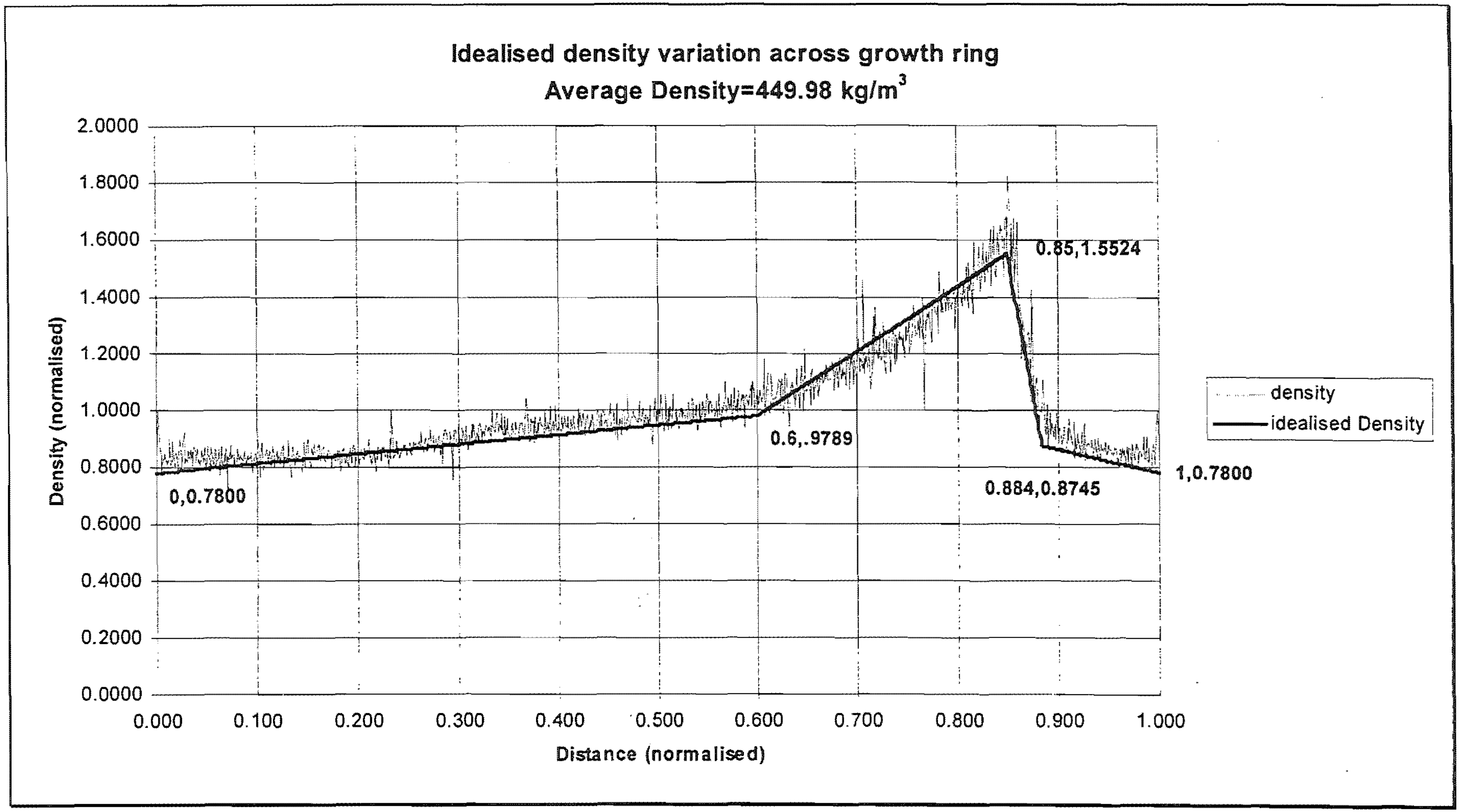


Figure 2.9 Piecewise approximation to density variation

2.4.3 Trend of radial diameter across growth ring

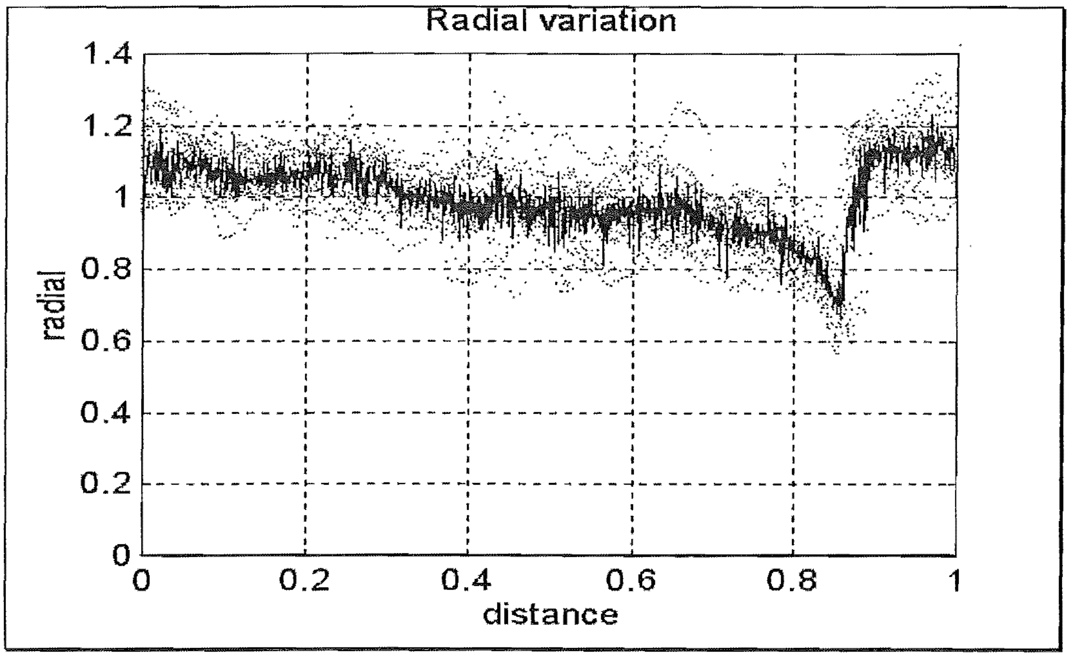


Figure 2.10. Plot of normalised radial diameter vs normalised distance

Radial diameter exhibits a gradual decreasing trend till the on-set of late wood where the rate of decrease of radial diameter increased markedly and reaches minimum at 0.85. A rapid increase in radial diameter is observed from 0.85-0.90 when the early-wood starts.

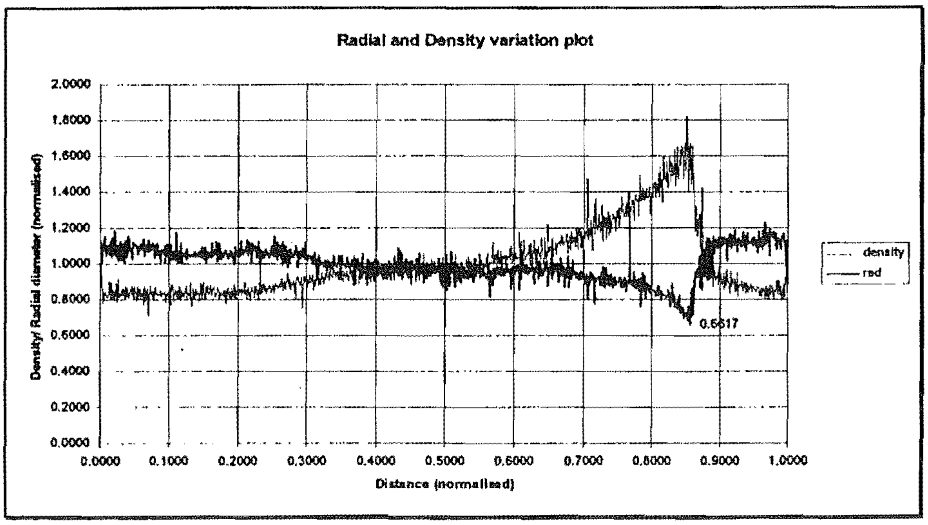


Figure 2.11. Normalised density and radial diameter plots

The interesting point is the transition between late-wood and early-wood characterized by a steep increased in radial diameter is mirrored strongly in the density variation. This is expected as density maximises at the point of thick cell wall and increased area ratio which coincides with low radial diameter late-wood.

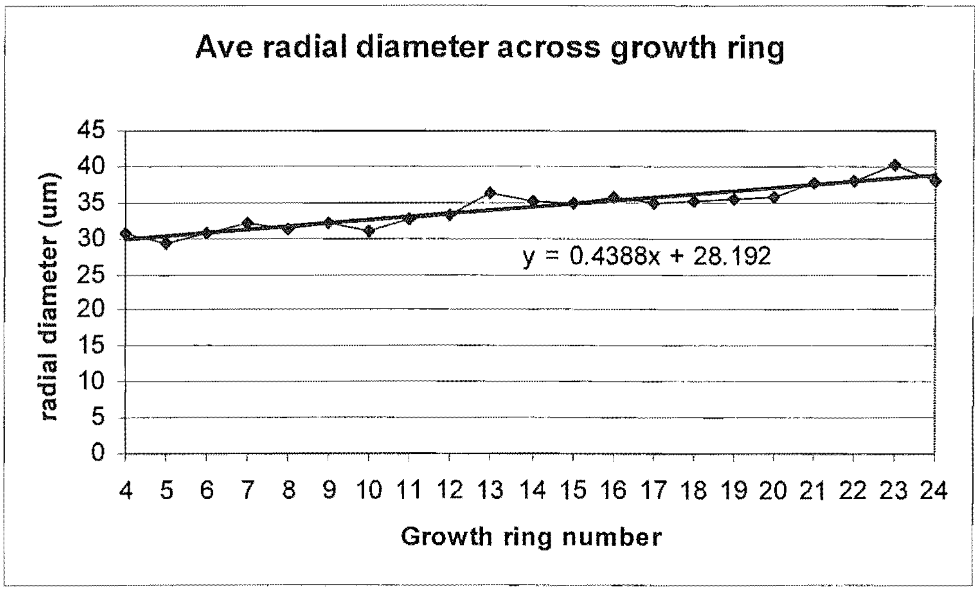


Figure 2.12 Variation of radial diameter across growth rings.

Variation of the average radial diameter across growth rings shows a linear trend, with the average radial diameter increasing linearly at a rate of 0.44 μm per year. This trend is specific to this sample as the radial diameter growth is very sensitive to external factor such as rainfall, nutrient level and other external factors that are beyond the scope of this study.

The normalized radial diameter is represented by piecewise-linear approximation in figure 2.13.

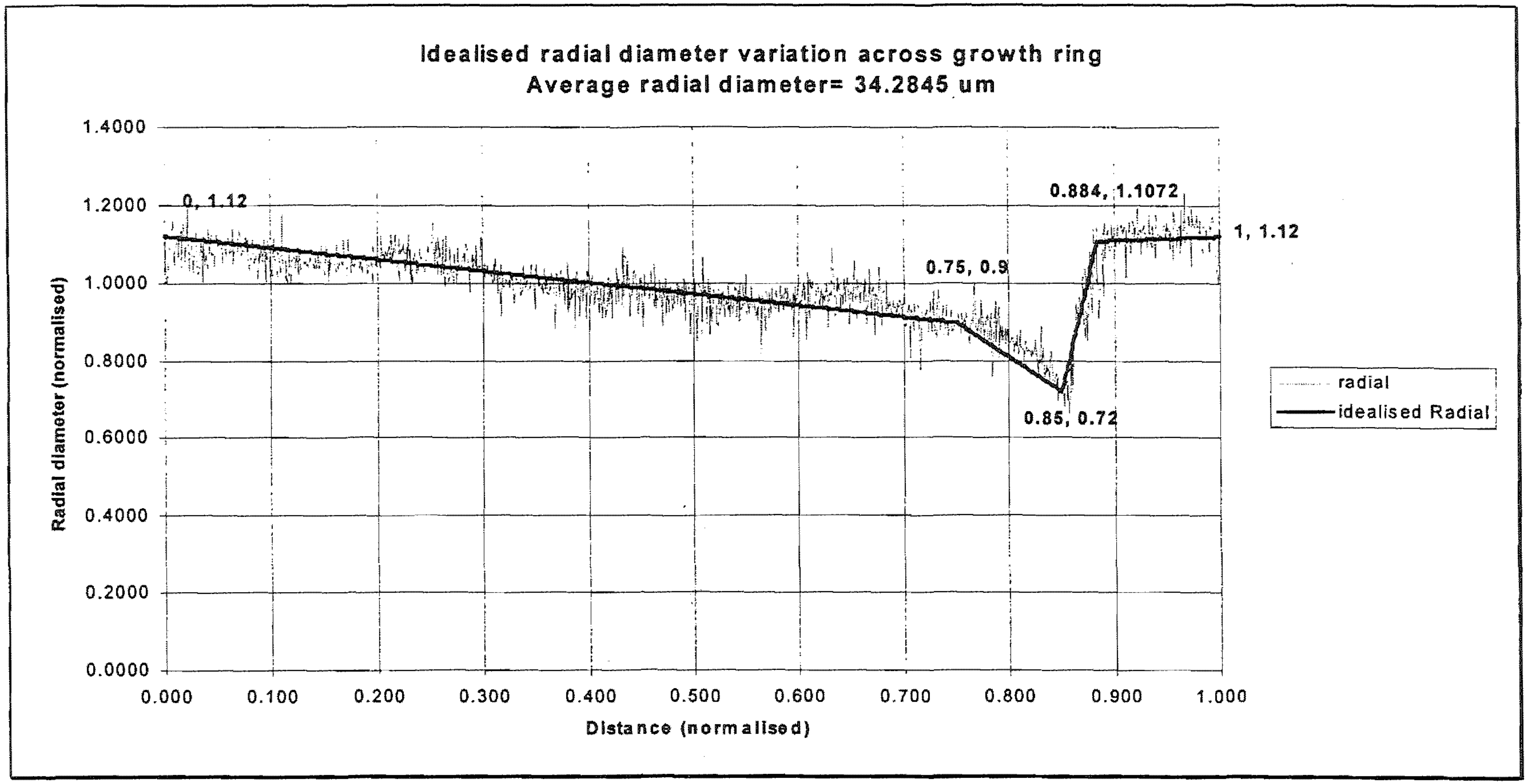


Figure 2.13. Piecewise variation of radial diameter across growth ring, idealized.

2.4.4 Trend of tangential diameter across growth ring

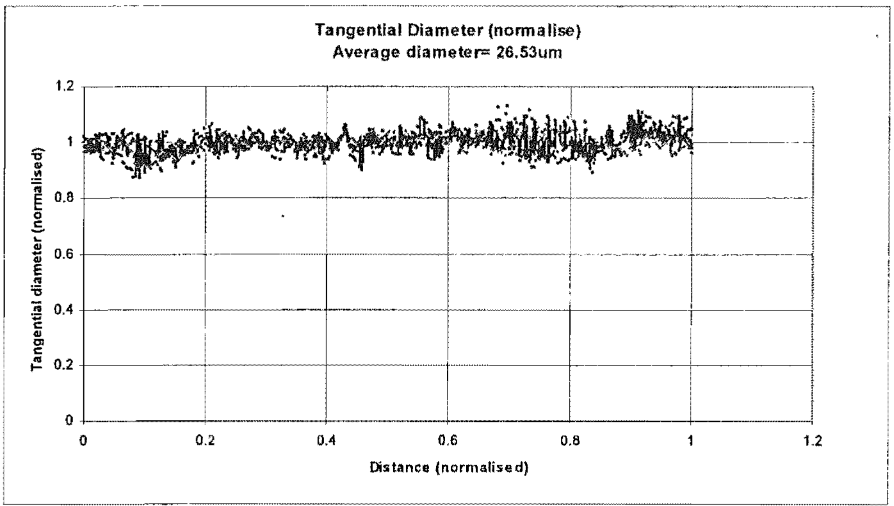


Figure 2.14. Tangential diameter vs Distance

Tangential diameter is relatively constant over the entire growth ring. This is consistent with observations and the growth mechanisms of the wood cells in the microscopic level. The fitted linear line is slightly higher than the observed average value (1.01) and is taken as unity in the model.

The variation of tangential diameter across growth rings show a very gradual increase in diameter of 0.23 μm per year.

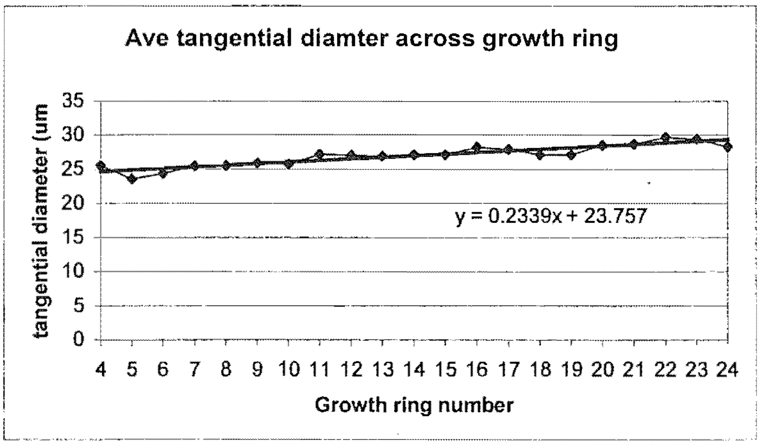


Figure 2.15 Variation of average tangential diameter with growth rings

2.4.5 Trend of microfibril angle (MFA) across growth ring

As explained earlier, two methods of MFA measurement are under considerations, the X-ray method and the Stain-Cell method.

3 growth rings are compared because Stain-cell MFA values are available from these 3 rings because of the limitations of this method.

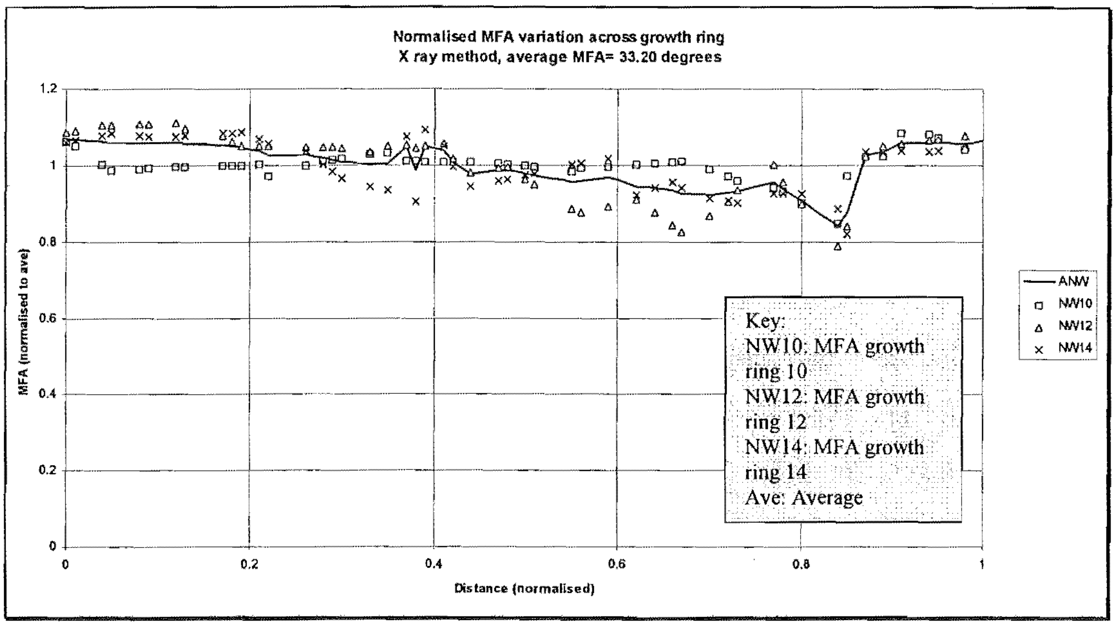


Figure 2.16. MFA (W values) vs distance

The X-ray method shows a clearly defined trend, MFA decreasing from early wood to late-wood and increasing again at the end of the late-wood period. The concern with the X-ray method is the acceptance of the W values as an indication of the average MFA values. The averaging effect and the resolution of the X-ray beam are primary concerns. The X-ray beam with a 0.5mm by 0.5mm cross section area would result in lower MFA values for late-wood because the X-ray beam would inevitably cover some early wood cells in the process of measurement (late-wood is estimated to be 10%). However the X-ray technique is a good rapid technique to obtain average MFA information.

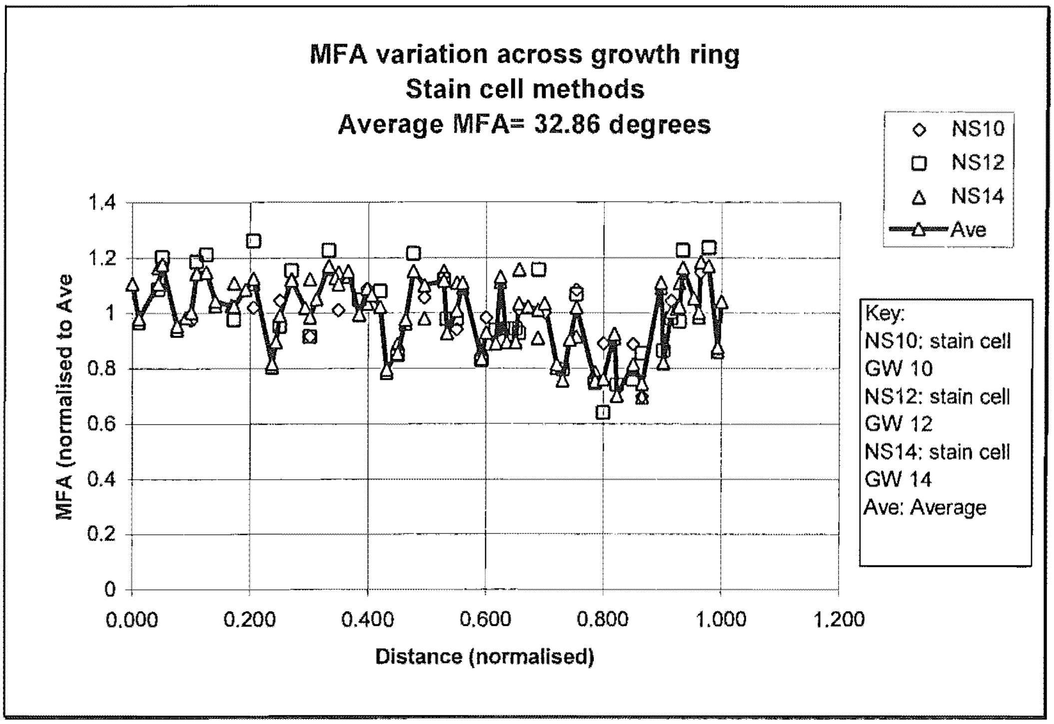
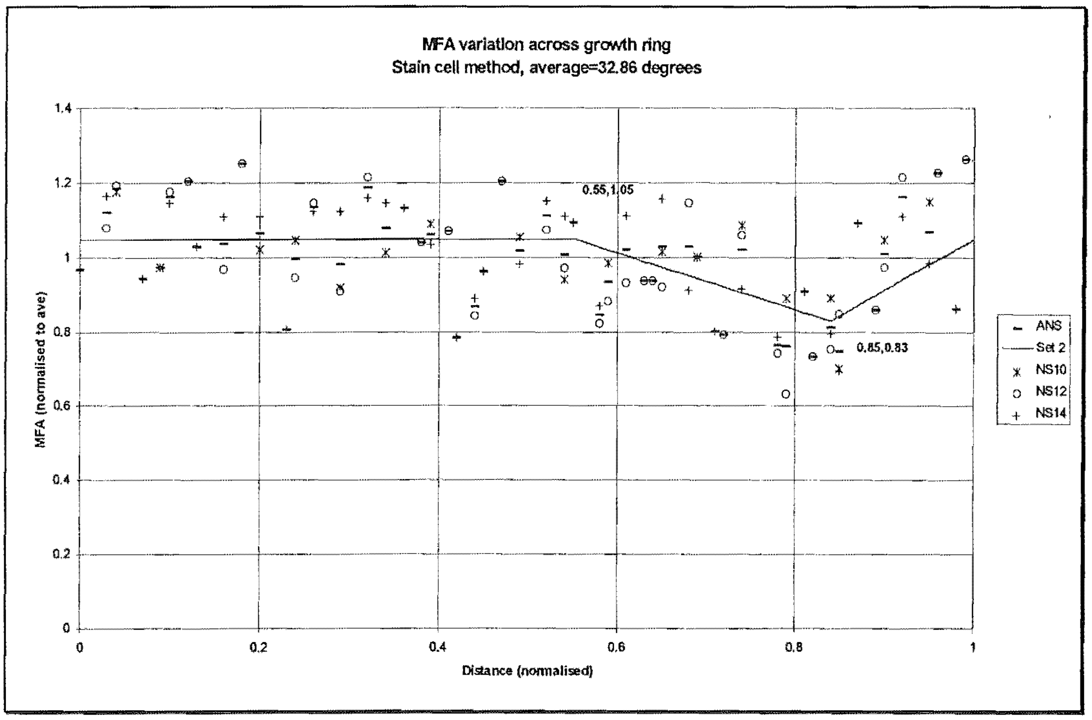


Figure 2.17. Stain Cell values vs distance

Stain-Cell values are direct measurements and highly localized. Unfortunately the values obtained are highly scattered. These values are strongly influenced by the choice of the sites during measurement as MFA values are not constant across a cell wall (speculated to be normally distributed about a mean value) and affected by local presence of imperfections in the cell wall such as pits. The variations are not limited to between tracheids but within a single cell wall of a tracheid. The selection of the point of measurement has a huge influence the readings.

Both methods exhibit the relatively similar trends and the lowest point occurs at a normalized distance of 0.85, consistent with other trends.

Both MFA trends are idealized as FEM inputs in figure 2.18 and 2.19.



Key:

ANS: Average normalised Stain cell

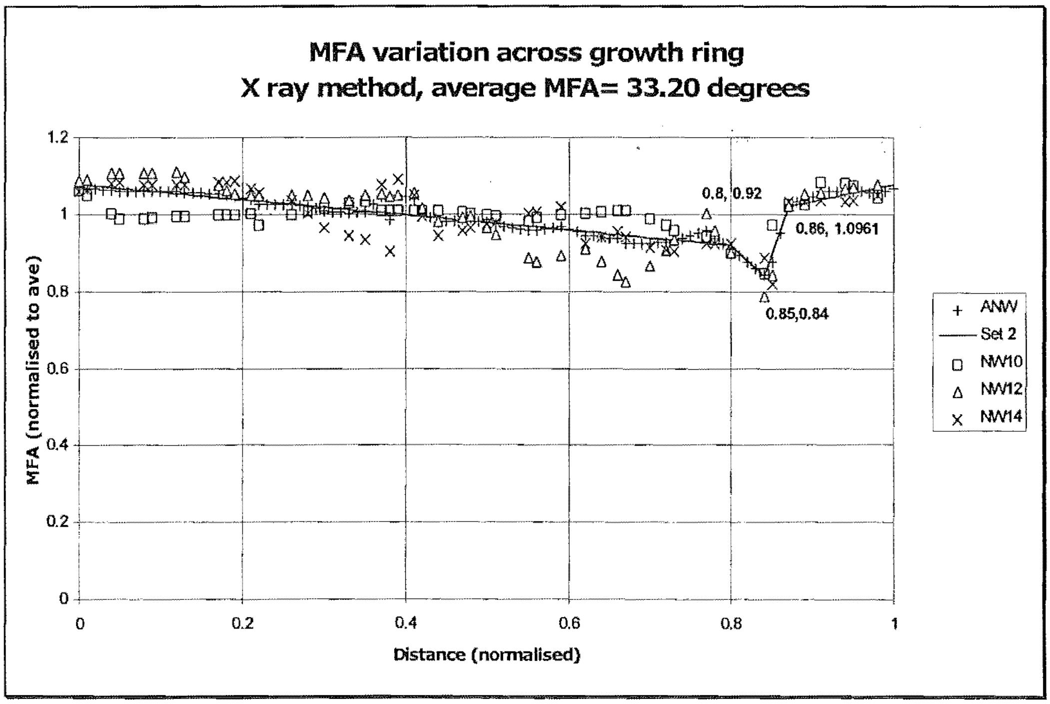
Set 2: Idealised Stain-cell approximation

NS10: Normalised Growth ring 10

NS12: Normalised Growth ring 12

NS14:

Figure 2.18. MFA Stain-Cell method piecewise approximation



Key:

ANW: Average normalised XRD cell

Set 2: Idealised XRD approximation

NS10: Normalised Growth ring 10

NS12: Normalised Growth ring 12

NS14:

Figure 2.19. MFA X-Ray method piecewise approximation

2.4.6 Trend of ring width across growth ring

The ring width is not a modelled parameter but included for completeness reason. The plot of ring width against ring number does not show any trend. The width of the ring formed is highly depend on the variables such as rainfall, nutrient level, peak temperature, hours of available sunshine, disease level and possibly a host of other variables that are not completely understood without a control experiment, however the broad trend is ring width tends to decrease with ring number.

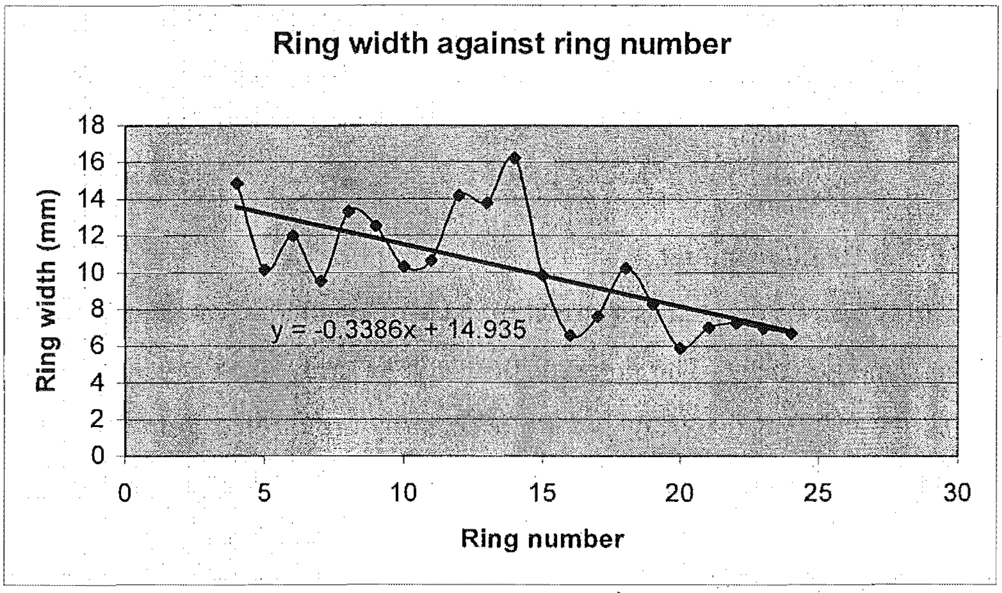


Figure 2.20. Plot of ring width against Growth ring number.

Chapter 3

Extension of Regular Model to Growth Ring Model

3.1 ANSYS Finite Element Model

The program used in the modeling of FE model is ANSYS 52. This program [50] supports macro for generation of model, constraints and 100 layered thick shell, SHELL91 element with in-plane strains and shear deformation. User defined tables and calculations, interpolation, auto scaling and translation build-in macros are necessary features for simplification of the macros required for the generation of growth ring model.

3.2 Single Cell Regular model

A regular array may be defined as a structure made of smaller repeated units, each unit having the same geometry. The properties of such an array need not be transversely isotropic in the plane of the array, but are necessarily orthotropic from the symmetry of each unit.

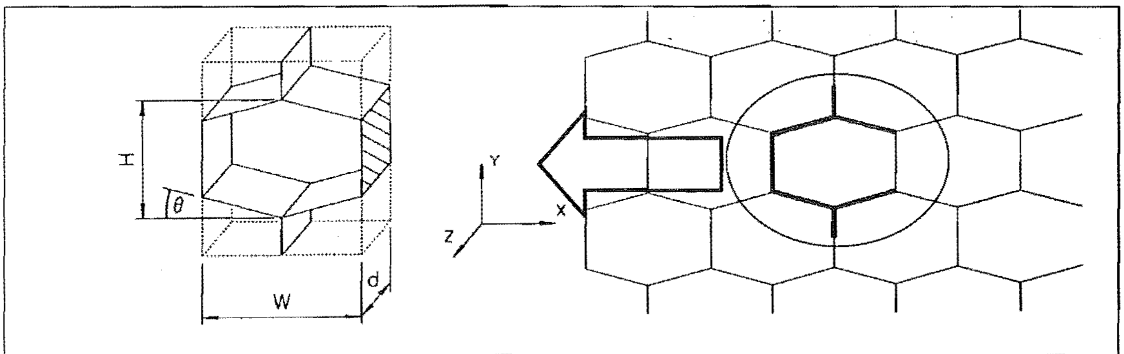


Figure 3.1 Regular model as a representation of a regular hexagonal array

By extruding such a repeated unit, a prismatic three-dimensional regular model can be define in the X, Y, Z co-ordinate system of ANSYS; X axis being co-related to the radial direction, Y axis being the tangential and the Z axis being the longitudinal direction within the wood microstructure co-ordinate system.

Parameters defining the model's basic geometry include; cell height, H; cell width, W; cell wall angle, θ; depth, d; and wall thickness, t. Note that cell heights are determined from the middle of the cell wall to the middle of the opposite cell wall and are not lumen dimensions. Other parameters which quantifies the elastic property of the model includes cell wall area ratio, r and the density of the wall material (assumed as 1460 kg/m^3). The relationship of area ratio r to the rest of the measurable parameter is given by:

$$
\begin{aligned}
r &= \frac{\text{total cell wall area of cross - section}}{\text{total area of the cross - section}} \\
&= \frac{t(H\cos\theta - W\sin\theta + W)}{W\left(H\cos\theta - \frac{W}{2}\sin\theta\right)}
\end{aligned}
\tag{3.1}
$$

A thick laminated shell (SHELL 91 element) which incorporates the effects of transverse shear, bending and membrane deformations represents the cell wall. The composition and make-up of the cell walls is discussed earlier in section Chapter 1.3.

The modeling of early wood by FEM was restricted to single regular hexagonal cell model for parametric studies in earlier work [2] mainly to establish a suitable geometrical shape to represent the typical wood cells. The approach of a Regular model to approximate the typical wood micro-structure has the obvious advantage of ease of simulation compared to the Irregular model because geometrical parameters necessary to complete the Regular model are easily available from Silviscan information or published work.

Earlier result [48] with statistical analysis of wood cell micro-structure and parametric studies of Regular model shows consistency with measured elastic modulus properties of similar softwood species at low MFA.

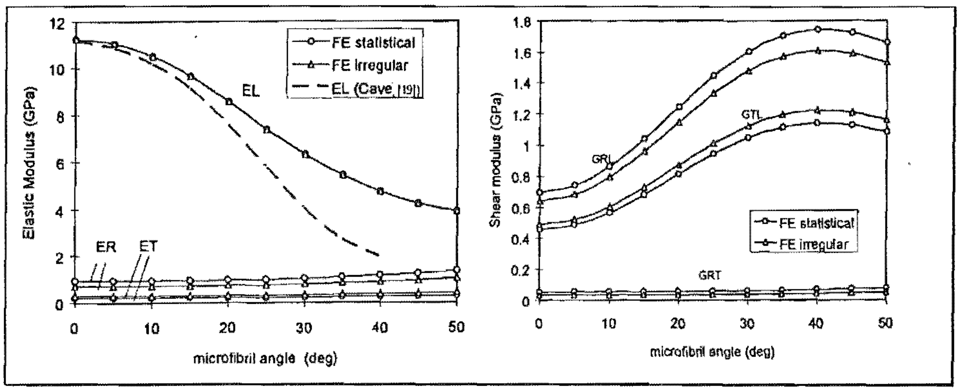


Figure 3.2 Calculated elastic modulus of FE Statistical and Irregular models [2]

	E_L	E_R	E_T	G_{LT}	G_{LR}	G_{TR}	ν_{TR}	ν_{LR}	ν_{LT}
FE-statistical (μ=0°)	11.26	.94	.22	.46	.70	.055	.36	.25	.25
(μ=10°)	10.52	.94	.22	.56	..86	.055	.36	.31	.31
FE-irregular (μ=0°)	11.26	.72	.31	.49	.64	.034	.46	.25	.25
(μ=10°)	10.51	.73	.31	.60	.79	.035	.46	.31	.31
Sitka spruce [27]	11.6	.90	.50	.72	.75	.039	.25	.37	.47
Norway spruce [27]	10.7	.71	.43	.62	.50	.023	.31	.38	.51

Table 3.1 Computed and measured orthotropic elastic constants (Basic density 390 kg/m^3, moisture content 12%, area ratio 0.267, cell height 23.02 μm, cell width 26.38 μm, cell angle 15.9°)[2].

These earlier models by Stol [2] used Regular six-sided hexagon models with no-offset, a refinement of the model to account for the anisotropic ratio of E_R/E_T with an off-set α of 0.65 [10] brings the ratio closer to a published [29] value of 1.7.

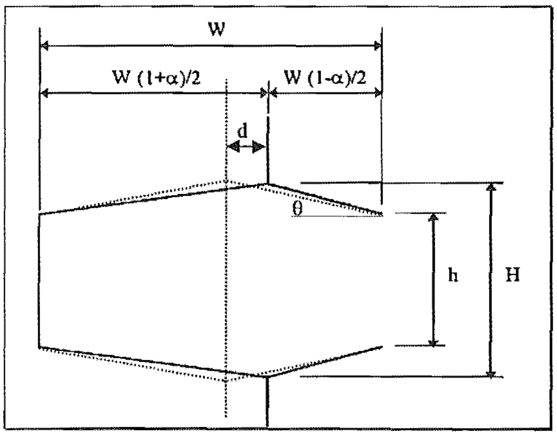


Figure 3.3 Regular off-set model

For the off-set model, the wall thickness is modified by the off-set factor α.

$$t = \frac{rW(H+h)}{2h+\sqrt{(H-h)^2+W(1-\alpha)^2}+\sqrt{(H-h)^2+W(1-\alpha)^2}} \tag{3.2}$$

where h is the tangential wall height. The depth of the model is calculated to give the element an aspect ratio as close to one as possible to reduce poorly shaped elements.

The model are constructed with SHELL 91 elements and verified that single element per cell wall is accurate within 1.5% of the analytical solution for isotropic material model.

3.2.1 Cyclic Constraints

Cyclic constraints relationships between nodal degrees of freedom must be imposed for isolated repeated unit to behave as in the continuous structure which it models. Such constraints are necessary in the analysis of the regular model, compatibility require that any face of a unit must deform to the same shape as the face directly opposite, while permitting an overall state of uniform normal or shear strain to exist in the bulk structure.

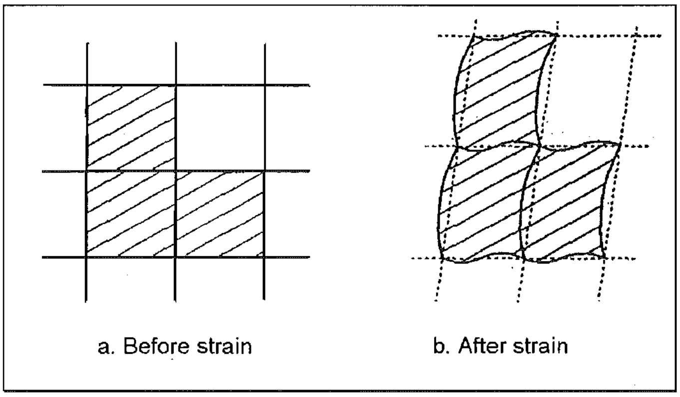


Figure 3.4 Cyclic constraints

A detailed mathematical treatment of cyclic constraints and specific constraints for each load case to determine the material properties is found in reference [2].

The equivalent homogenous elastic properties are obtained by loading the FE model to simulate direct shear and expansional loadings in the L-R-T direction. The computed deformations of the model then determined the moduli and expansional coefficients for an equivalent homogeneous material governed by the orthotropic stress-strain relationship of equation 1.2.

3.3 Growth-ring regular model

To enable the model to co-relate to macroscopic elastic properties, growth ring's geometrical variations need to be considered within the FE model. The basic Single cell regular model with offset is used as the primary cell model to model the Growth-ring.

To construct the growth ring model, each cell is constructed with the idealised geometrical information obtained from the Silviscan measurement. ANSYS 52 has capability to interpolate between points from a table of the co-ordinates of the idealized trends. Preliminary data from growth ring 14 are used primarily because experimental measurement [11] are being carried out on growth ring 14.

3.3.1 Algorithm of Growth ring model

The idealized geometrical parameters in section 2 are normalised to the average value of each respective parameter. To enable these parameters to be used to reconstruct an idealised Growth Ring 14, the average geometrical parameters for Growth Ring 14 are used.

Cell Width = 34.28 μm
Cell Height = 26.53 μm
Average MFA = 24.7°
Average density = 445 kg/m^3
Cell angle = 15.89°
Actual Growth ring width = 15.5 mm
Model ring width = 1885 μm

Approximately 450 cells are required to completely reconstruct an idealised Growth ring 14, a model of such complexity and size will demand computing power that may be beyond the resources of some research facilities. To simplify this requirement, a convergence test is performed in which the number of cells used to model Growth ring 14 is gradually increased. The solutions are plotted and compared to the asymptotic values; the number of cells which gives a deviation of less than 1.5% from the asymptotic value is selected as a reasonable compromised to model the mechanical properties of a growth ring; 55 cells being used in this case.

By changing the growth-ring width to 1885 μm, the equivalence of 55 wood cells, we can control the size of the model. The construction of the growth ring begins with the input of necessary geometrical information of the growth-ring 14 width, cell width, cell height, Average MFA, cell angle and average growth-ring density. The first cell is constructed with the width, height, density and MFA being computed from the respective normalised parameter set at a normalised distance of zero, cell angle being fixed at 15.89° for this Growth ring. Cell wall tangential height h is computed by the geometry in Figure 3.1.

$$h = H - W_1 \tan\theta \qquad (3.3)$$

$$r_i = den_i / 1460 \qquad (3.4)$$

where den_l (density) and r_l denotes the properties of the ith cells, H and W_l denotes the first cell. The tangential height h is constant for all cells for proper connection of element nodes with the adjacent cells and to preserve the constant cellular height, H. The thickness of the cell wall t_i is calculated using equation (3.2). The cumulative centroidal distance of the cells is normalised to 1885μm and the $i+1$th cell width, density and MFA is then calculated. The $i+1$th cell is constructed by linear stretching the first cell by a factor of W_i / W_1 in the radial direction and translating the stretched cell to the end co-ordinates of the ith cell. The model construction is terminated when the cumulative cell radial width just exceeds the defined ring width; 1885 μm in this case.

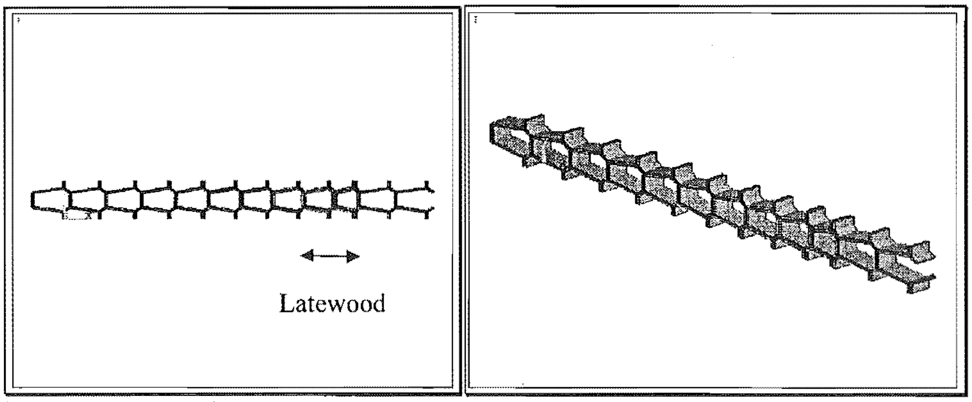


Figure 3.4. Growth ring regular model with 12 wood cells. Note the late-wood cells with small widths and thick walls.

3.3.2 Convergence test

A convergence test was performed to test the convergence of the model in relation to the number of cells to represent an entire 450-cell growth ring 14 and it was found that a 55-cell model is sufficient in getting within 1.5 % of the steady state solution.

A plot of number of cell versus percentage error to the asymptotic values is shown in figure 3.5.

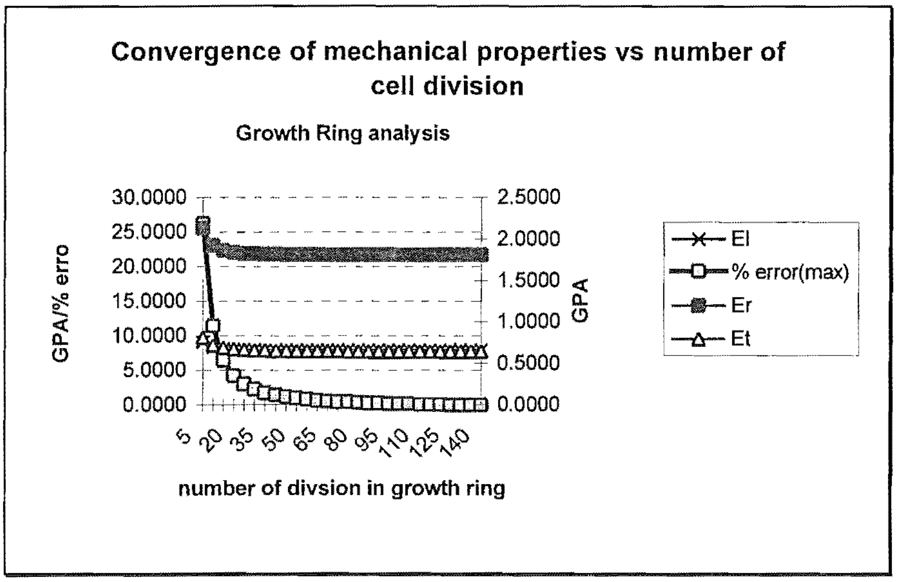


Figure 3.5. Convergence of growth ring model

3.4 Comparison of Regular and skeletal Early wood, middle wood and latewood models

Growth ring 14 is chosen because the reference samples are prepared from this growth ring for both experimental measurement, micrography and FE modeling. The growth ring is modeled using both Silviscan and Stain-cell MFA values, and a uniformly distributed off-set model using the Gauss Legendre [10] integration method.

Thin-section micrographs of Growth ring 14 are prepared in three sections, early-wood, middle-wood and late-wood according to the locations with-in the growth ring 14. The micrographs are then used to prepare Skeletal FE models

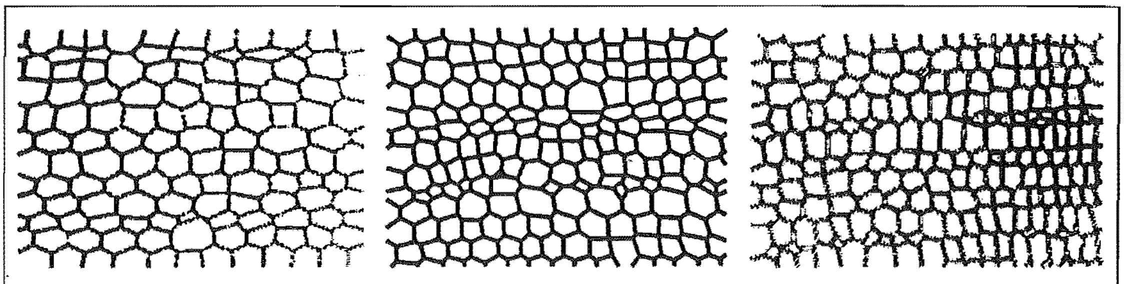


figure 3.6 Skeletonised models generated from micrographs

The Regular models based on the same data have also been generated. The geometrical parameters which define a regular model are illustrated in figure 3.3. They include the maximum radial cell width W and cell height H, and the shape parameters d and θ which determine the squareness and alignment of the cells. Values for W and H are obtained either from measured data from Silviscan or by direct measurement from the micrograph images. The parameters d and θ are more difficult to relate to measured or measurable data. The first determines the extent to which the tangential walls are "staggered" in adjacent radial ranks of cells; the second defines the alignment and orientation of the radial walls. Both are important in determining the transverse stiffness of the material. When $\theta = 0$, for example, the radial walls are perfectly aligned giving an uncharacteristically high elastic modulus in the radial direction. As θ approaches 30° - and for the case of cells whose walls are of equal length - the model becomes transversely isotropic. This is equally uncharacteristic of softwood for which the ratio of radial to tangential stiffness is typically of the order of 2:1. An appropriate value of θ for modeling purposes clearly lies somewhere between these extremes. In the current study, a value of $\theta = 15.89°$ is used throughout. In the cellular model of Persson [26] a similar value appears to have been used.

The second shape parameter, d, has a strong effect on the transverse modulus E_T. A non-dimensional parameter a defined by $\alpha = 2d/W$ has the property that $\alpha = 0$ and $\alpha = 1.0$ give maximum and minimum staggering of the tangential walls. Cell micrographs indicate that most values of α appear to be present in the cell geometries typical of radiata pine. It is difficult to accommodate this variability within a regular cellular model. In the current model, it has been partially achieved through a further homogenisation in which a uniform distribution of α is assumed to occur between layers of cells orientated in the R direction. The laminate homogenisation method of Chou and Carlene [24] is used to obtain equivalent homogenized quantifies. Results demonstrate that this additional modification brings the computed results obtained from regular models closer to those obtained from the more disordered arrangement characteristic of skeletal cell micrographs.

Computed results for the stiffness of earlywood, middle-wood and latewood obtained from the three models shown in figure 3.6 are presented table 3.2. Also shown are results obtained from two regular models; a "standard" model with offset α=0 (denoted by Reg (b)) and a homogenised version (denoted by Reg(a))-in which α is uniformly distributed over the interval $0<\alpha<1$. The correspondence between skeletal and regular models is generally good except for the transverse moduli E_R and E_T which are particularly sensitive to the transverse geometry. The homogenised regular model (Reg(a)) is somewhat better representing these parameters.

Although the correspondence between the regular homogenised and skeletal models is by no means perfect, the regular model has considerable practical appeal since it does not require the capture of cell micrographs to generate model geometry. This is a significant practical limitation on the use the skeletal model. The regular homogenised model requires only readily measurable parameters as input - mean cell dimensions, MFA and basic density-and can readily be applied to study the effects of variations of these parameters within and between trees. The remaining results are obtained from the regular homogenised model or from an equivalent homogenised version of the regular growth ring model.

	Earlywood			Middle-wood			Late wood		
	Skel	*Reg(a)*	*Reg(b)*	*Skel.*	*Reg(a)*	*Reg(b)*	*Skel*	*Reg(a)*	*Reg (b)*
E_L	6.83	*6.85*	6.84	8.40	8.41	8.41	13.44	13.48	13.47
E_T	0.22	0.30	0.16	0.42	0.57	0.32	1.34	1.12	0.70
E_R	0.56	0.88	0.91	0.80	1.31	1.39	1.11	2.10	2.30
ν_{TR}	0.47	0.40	0.31	0.48	0.39	0.32	*0.56*	0.36	0.31
ν_{LR}	0.49	0.49	0.49	0.49	0.49	0.49	*0.45*	0.45	0.45
ν_{LT}	0.49	0.49	0.49	0.49	0.49	0.49	0.45	0.45	0.45
G_{TR}	0.04	0.04	0.05	0.07	0.07	0.08	0.12	0.14	0.16
G_{LT}	0.87	0.86	0.80	1.16	1.17	1.09	1.79	1.53	1.69
G_{LR}	1.32	1.36	1.41	1.65	1.65	1.71	1.57	1.97	1.82

Table 3.2. Elastic constants for earlywood, middle-wood and latewood. (all moduli in Gpa) [3]

3.5 Measured and computed results for stiffness.[5]

Measured values of shear and elastic moduli have been obtained for test specimens taken from the growth ring shown in figure 3.7. The procedures used to obtain these data are reported by Winkelmann [4]. The number of test specimens within the growth ring is dependent upon the specific elastic parameter being measured. Measurements of the axial modulus E_L and shear modulus G_{LT} are made on fourteen equally spaced specimens through the growth ring; measurements of G_{LR} are made on six equally spaced specimens and measurements of E_T and G_{RT} on four. The measurement technique used for E_R is such that the value obtained for seven separate specimens relates to the growth ring as a whole rather than to a specific radial location.

Averaged values of mean MFA, density and cell size for each specimen are inferred from its location within the growth ring and from the data of Chapter 2. These values were used to construct a regular FE model for each test specimen. The orthotropic moduli for the growth ring as a whole have also been calculated by using the growth ring model of section 3.3.

Comparisons of computed and measured values for the longitudinal, radial and tangential Young's moduli E_L, E_T and E_R are presented in figures 3.7. The ordinate in each case is the sample identification number. In the case of E_L and E_T (figure 3.7a and b) this forms in effect a scaled radial coordinate since the test specimens are

[5] Article from Article from IUFRO conference 'Microfibril angle in Wood' at Westport, NZ 1998 [28], co-authored by author.

spaced at roughly equal intervals through the growth ring, the first and last data points in each case corresponding to late-wood. In the case of E_R (figure 3.7c) the test specimens contain material from the whole growth ring. The sample number then has no significance in terms of radial location and the measured values of E_R are compared to the computed "growth-ring" value obtained from the cellular model of section 3.3.

The trends in the variation of the measured data for E_L and E_T through the growth ring are reflected in the computed values although the late-wood effect is somewhat overestimated. The correspondence between the computed and measured values of E_R for the growth ring as a whole (figure 3.7c) is close.

Comparisons of computed and measured values of the three orthotropic shear moduli G_{LT}, G_{LR} and G_{RT} are shown in figure 3.7d-f. Here the observed variation through the growth ring is reflected in a qualitative sense by the computed values although those for G_{LT} and G_{LR} - particularly the former - are significantly higher than the measured values. The reason for this discrepancy is not clear. It may be associated with deficiencies in the measurement method -which is indirect - or with real physical effects not represented adequately within the model (the stiffness-reducing effect of border pits in the radially aligned walls for example).

It is difficult from the current measured data to isolate the effects of MFA, density and cell dimensions in determining the elastic moduli since all of these parameters vary simultaneously between samples. Similar measured data now being obtained for growth rings closer to and further from the pith may help in this regard. The effects of MFA, density and other micro-structural characteristics can however be isolated quite simply by using the computer model.

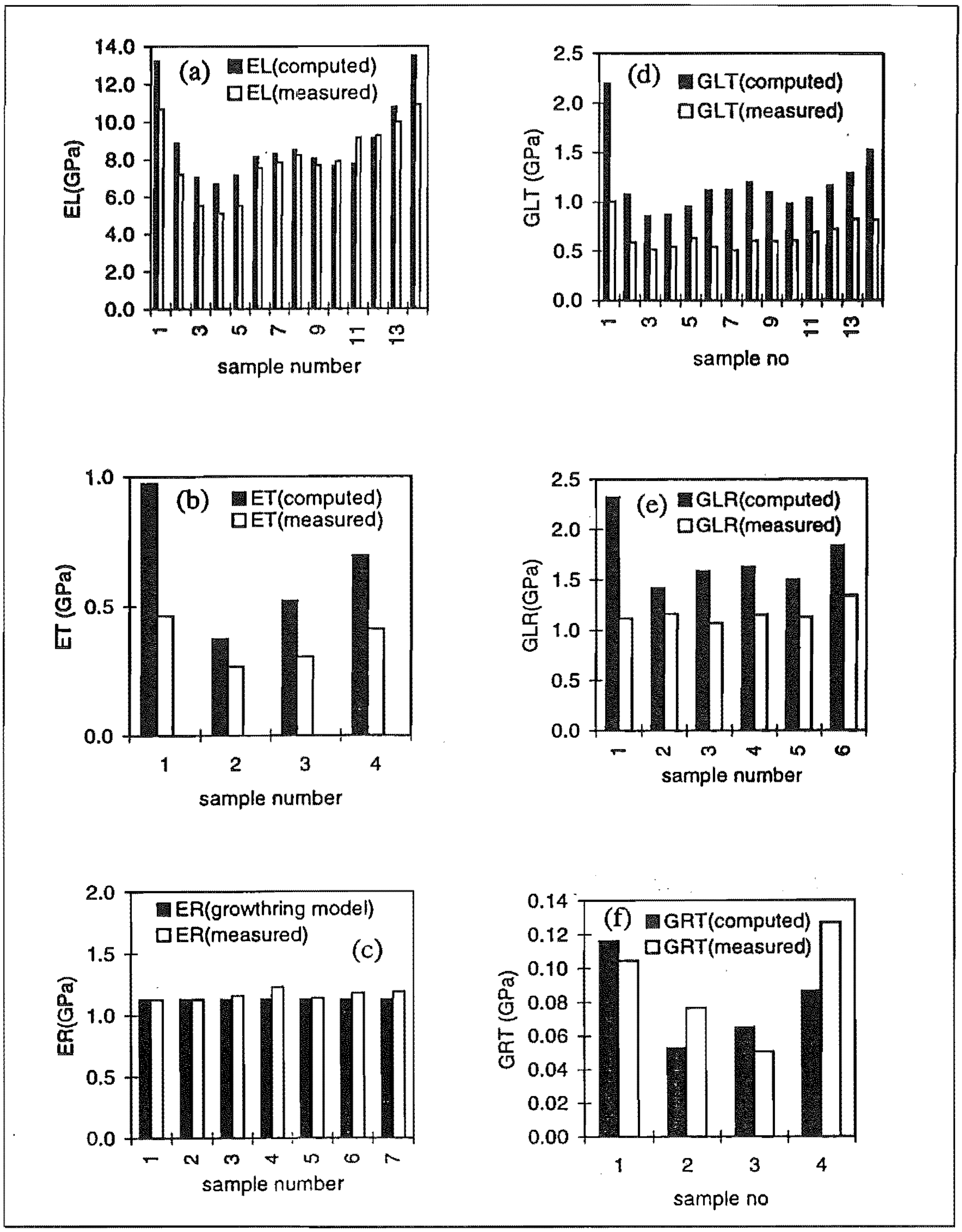


Figure 3.7 Computed and measured Elastic moduli and shear moduli
Through a growth ring (growth ring 14)

3.6 Predicted variations of stiffness with MFA

The cellular FE model has been used to predict the variation of macroscopic elastic moduli with MFA for a range of densities typical of plantation grown Radiata pine. The homogenised growth-ring model is used. The predicted values of the elastic moduli therefore correspond to macroscopic values at board scale. A moisture content of 12% is assumed. Results are presented in figure 3.8,a-f which show the six independent direct and shear moduli plotted against MFA for averaged basic densities in the range 300kg/m^3 to 500 kg/rn^3. The cell-wall lay-up and the relative proportion of the Si, S2, S3 and CML layers are held constant irrespective of density and MFA.

The variation of the axial Young's Moduli E_L with MFA is shown in figure 3.8a. This exhibits the same strong dependence demonstrated by the cell wall models of Cave and others and confirmed by measurement. The transverse moduli E_R and E (figures 3.8b and 3.8c) show a weaker variation with MFA as does the transverse shear modulus G_{RT}. (figure 3.8d) whereas the longitudinal shear moduli G_{RL} and G_{TL} (figures 3.8e and 3.8f) are strongly dependent, doubling or trebling in value as MFA increases from values characteristic of the outerwood (10°~15°) to those characteristic of corewood (40°~50°). This behaviour is not unexpected given that the shear stiffness of the cell wall in its own plane increases to a maximum as the microfibril angle approaches 45°. All of the trends illustrated in figures 3.8, a-f closely parallel similar variations predicted by Persson [26] for spruce.

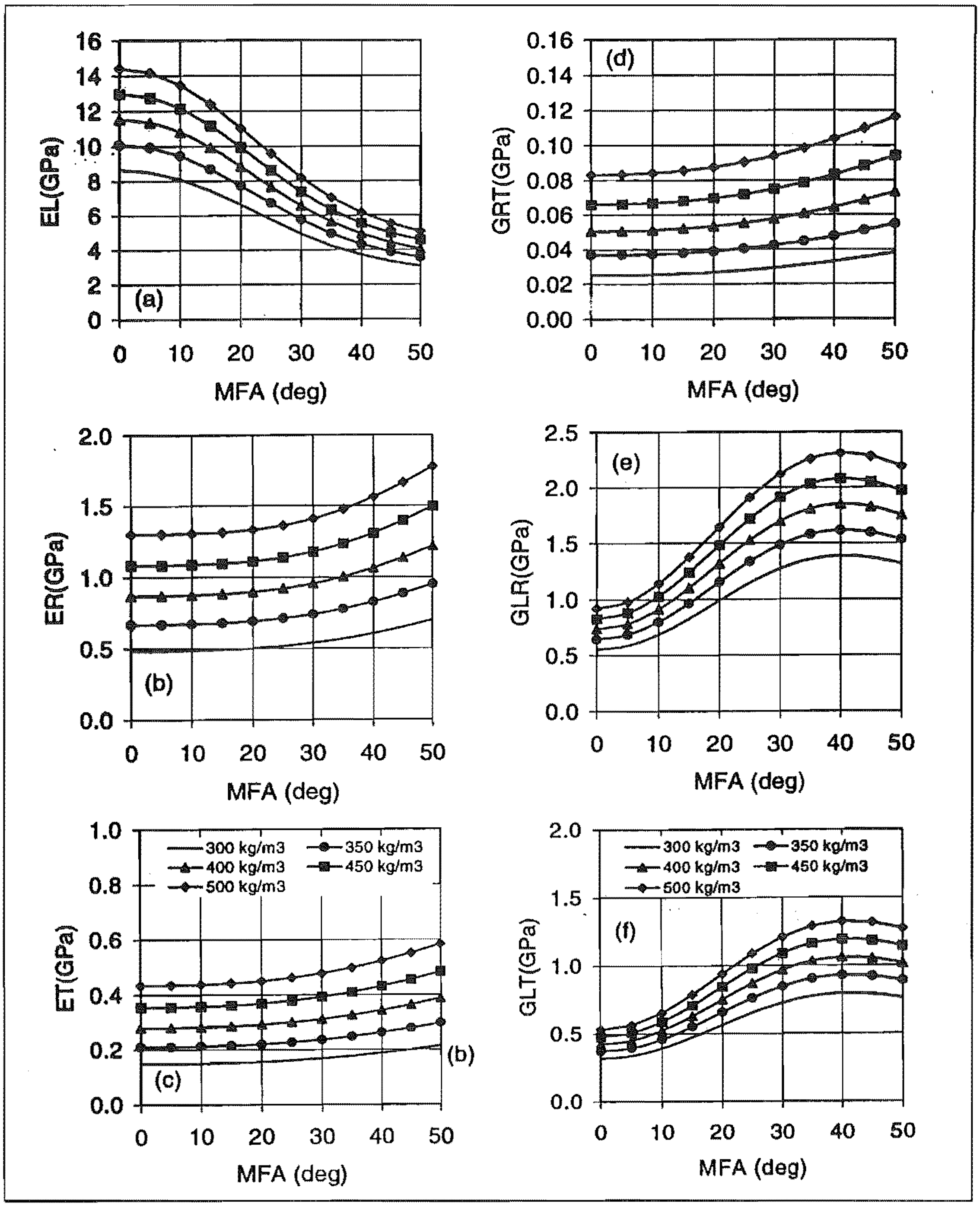


Figure 3.8 Variation of Moduli versus MFA. Growth ring model.

Chapter 4

Introduction to Energy and Finite Element method

4.1 Introduction to Energy method (Rayleigh-Ritz)

The energy principle is that of 'virtual work' and is widely used in numerical methods for solving complex engineering problem. Finite element method is a systematic application of the Rayleigh-Ritz method, at least for the application of FEM in the field of static stress and strain analysis. When stated for a single particle, it appears trivial, but applied over a system of particles, it is significant and a useful basis of analysis. A finite incremental displacement which does not alter the forces acting on a particle is termed 'virtual displacement' and becomes the principle of virtual work. Briefly stated,

> *A necessary and sufficient condition for the equilibrium of a particle is the work done by the forces on the particle is zero for any virtual displacment.*

or $$\Delta W = 0$$

$$= \Delta W^{\text{int}} + \Delta W^{\text{ext}} = 0 \tag{4.1}$$

where $$\Delta W^{\text{int}} = \sum_{i=1}^{n} \Delta W^{\text{int}} \tag{4.2}$$

and $$\Delta W^{ext} = \sum_{i=1}^{n} \Delta W^{ext} \tag{4.3}$$

4.2 Principle of total potential

Energy is defined as the capacity to do work. As work is done by the system, energy is lost. The internal energy of a solid body is the strain energy, negative being work done by the internal forces. Therefore the strain energy increment ΔU is given by

$$\Delta U = -\Delta W^{\text{int}} \tag{4.4}$$

By using equation (4.1),

$$\Delta U = \Delta\, W^{\text{ext}} \tag{4.5}$$

For external force, a decrement in energy $-\Delta V$ is defined as the work done by the external forcees during an incremental virtual displacement.

$$-\Delta V = \Delta W^{ext} \tag{4.6}$$

By eliminating ΔW^{ext}, we obtained

$$\Delta U = -\Delta V \quad \text{or} \quad \Delta (U + V) = 0 \tag{4.7}$$

The term $(U + V)$ is the 'total energy' of the system denoted by χ.

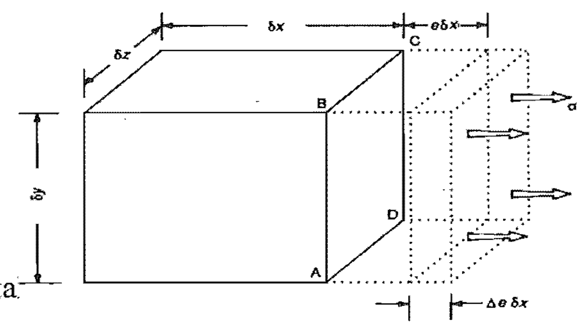
Equation 4.7 states that the total energy χ is zero during any virtual displacement or that no energy is dissipated from a 'closed' system- conservation of total energy. Internal elastic forces are conservative as they are recovered when strained bodies are returned to their original state. External forces of fixed magnitude are conservative as no net work is done during movement within a closed path. Such condition applies to an elastic system and holds true if displacements are infinitesimal small as in static problems.

4.3 Strain energy in an elastic continuum

The strain energy of an elastic continuum can be evaluated the energy expended during the process of deformation per unit volume (strain energy density). The system energy can be computed by integrating the strain energy density over the entire volume of the continuum.

4.3.1 Strain energy of direct stresses

Consider the case of normal stress σ acting on a body in the x direction. The resulting strain in the x direction is denoted by e.



The incrementa ... stress

$$\Delta U = (\sigma\Delta e)\delta^x\delta^y\delta^z$$
$$= (\sigma\Delta e)\delta V \qquad (4.8)$$

where δV is the incremental volume. If the body deforms from a state of zero stain to a final value of e', the strain energy per volume is

$$SE/vol = \int_0^{e'} \sigma de \qquad (4.9)$$

For a isothermal case and linearly elastic,

$$SE/vol = \frac{1}{2}\sigma e \qquad (4.10)$$

4.3.2 Strain energy in shear

In pure shear, the strain energy is

$$\Delta U = (\tau\Delta\gamma)\delta V \qquad (4.11)$$

where τ, γ and $\Delta\gamma$ are shear stress, shear strain and incremental shear strain. The integration over the process yields

$$SE/vol = \int_0^{\gamma'} \tau d\gamma \qquad (4.12)$$

$$= \frac{1}{2}\tau\gamma \qquad (4.13)$$

4.3.3 Strain energy in compound stress

The previous derivation of pure shear and direct stress can be applied to any orthogonal case by summing the two contributions. In the most general three dimensional case defined by stress components σ_1, σ_2, σ_3, τ_{12}, τ_{13} and τ_{23} with the corresponding strains, equation (4.9) yields

$$SE/vol = \int (\sigma_1 de_1 + \sigma_2 de_2 + \sigma_3 de_3 + \tau_{12} d\gamma_{12} + \tau_{13} d\gamma_{13} + \tau_{23} d\gamma_{23}) \quad (4.14)$$

4.4 The Finite element approach

The finite element is developed using the application of 'virtual energy'. The direct stiffness approach is used in the analytical solution because of the simplicity and the ability to developed the approach by modifying the stiffness matrix with different consideration.

The following are brief steps of the FE analytical process,

1. Subdivision of the continuum into a finite number of simpler discrete elements.
2. Formulate the element properties using different stress-strain relationships.
3. Assemble the elements into a finite element model of the structure.
4. Apply arbitrary loads to the model.
5. Specify the boundary conditions on the model.
6. Solve the nodal displacements.
7. Extracts the model stresses and strains to compute the predicted Elastic modulii.

4.4.1 Element definition

Element definition has a big impact on the results in FEM. The definition determine the capability of element in handling various type of stresses. The element displacement field $[u]$ is expressed in polynomial functions as an approximation to the displacements within the two nodal points. This polynomial function $[N]$ is known as the interpolating matrix or shape matrix. In general,

$$[u] = [N^e]\,[d^e] \qquad (4.15)$$

where the superscript e denotes the elemental properties.

The strain is defined as the derivative of the displacement $e_x = du/dx$ and in matrix notation

$$[e] = [B^e][d^e] \qquad (4.16)$$

where $[B]^e$ contains the components of the derivatives of the shape functions.

Assuming Hooke's law holds, the equation (4.16) can be expressed as

$$[s] = [D]\,[B^e][d^e] \qquad (4.17)$$

where $[s]$ is the stress matrix comprising of the components of $[\sigma]$ and $[D]$ is the stress-strain matrix with components of $[E]$.

In matrix form, the strain energy equation (4.10) can be expressed as

$$U^e = 1/2\,[d^e]^T\,[K^e][d^e] \qquad (4.18)$$

where $[K^e]$ is the element stiffness matrix given by

$$[K^e] = \int_V ([B^e]^T [D]\ [B^e])\, dV \qquad (4.19)$$

The potential energy due to external body forces is

$$V^e = -\int [u]^T [g]\, dV \qquad (4.20)$$

where $[g]$ contains the body forces.

Substituting equation (4.15) yields

$$V^e = -[d^e]^T ([f_g{}^e] + [f_T{}^e]) \qquad (4.21)$$

where $[f_g{}^e] = \int_V [N^e]^T [g]\, dV$ and $[f_T{}^e]$ is due to thermal stress effect.

For point forces, assuming an isothermal condition, the equation (4.21) reduces to

$$V^e = -[d^e]^T [f_g{}^e] \qquad (4.22)$$

The final total energy equation for the element is

$$\chi = \Sigma^m_{e=1}\chi^e = 1/2\, [d]^T [K] [d] - [d]^T [f_g] \qquad (4.23)$$

4.4.2 Beam Element formulation and direct stiffness approach

4.4.2.1 Simple Beam

Consider a simple Bernoulli-Euler beam 'element' parallel to the x-axis with a length L and cross-section A and Young's modulus of E. In this formulation, we assume that no lateral load was carried by the beam element and its strain energy being derived entirely from axial tension and compression. At node 1, the displacement is w_1 and at node 2 the displacement is w_2, a cubic nodal displacement vector will be a good estimation of the deformation.

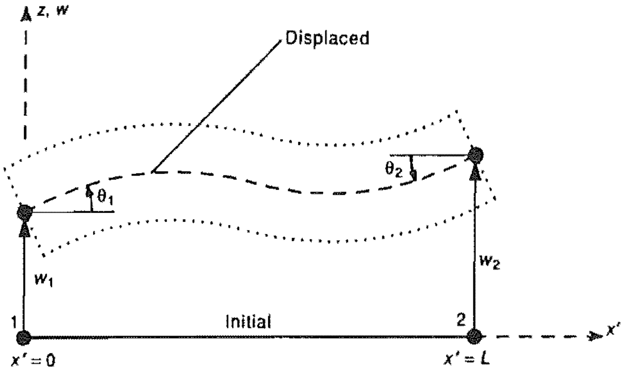


Figure 4.2 Simple beam element

$$W(x') = \alpha_1 + \alpha_2 x' + \alpha_3 x'^2 + \alpha_4 x'^3 \tag{4.24}$$

Consistent with this interpolation for w, the rotation of the neutral axis, $\theta(x')$ is given by

$$\theta(x') = dw/dx = \alpha_2 + 2\alpha_3 x' + 3\alpha_4 x'^2 \tag{4.25}$$

The constants $\alpha_1 \ldots \alpha_4$ are obtained by equating w and θ to nodal values at $x'=0$ and $x'=L$. After some manipulation,

$$\alpha_1 = w_1, \qquad \alpha_2 = \theta_1$$

$$\alpha_3 = -\frac{3}{L^2}w_1 - \frac{2}{L}\theta_1 + \frac{3}{L^3}w_2 - \frac{1}{L}\theta_2$$

$$\alpha_4 = \frac{2}{L^3}w_1 + \frac{1}{L^2}\theta_1 - \frac{2}{L^3}w_2 + \frac{1}{L^2}\theta_2 \tag{4.26}$$

When substituted into equation (4.16) yields

$$w(x') = n_1(x')w_1 + n_2(x')\theta_1 + n_3(x')w_2 + n_4(x')\theta_2$$

where

$$\begin{aligned} n_1(x') &= 1 - 3\left(\frac{x'}{L}\right)^2 + 2\left(\frac{x'}{L}\right)^3 \\ n_2(x') &= x'\left(1 - 2\left(\frac{x'}{L}\right) + \left(\frac{x'}{L}\right)^2\right) \\ n_3(x') &= 3\left(\frac{x'}{L}\right)^2 - 2\left(\frac{x'}{L}\right)^3 \\ n_4(x') &= x'\left(-\left(\frac{x'}{L}\right) + \left(\frac{x'}{L}\right)^2\right) \end{aligned} \tag{4.27}$$

The functions $n_i(1..4)$ are the shape components of the shape matrix $[N^e]$ of the element given in equation (4.15).

In a beam without axial tension and compression, the strain energy per unit length is given by

$$SE/length = \frac{1}{2}M\left(-\frac{d^2w}{dx^2}\right) \qquad (4.28)$$

where

$$M = EI\left(-\frac{d^2w}{dx^2}\right) \qquad (4.30)$$

Drawing its equivalence from equation (4.27), the curvature-displacement relationship gives

$$e = [B^e][d^e] \qquad (4.31)$$

where $$[B^e] = \left[-\frac{12x'}{L^3}+\frac{6}{L^2}, -\frac{6x'}{L^2}+\frac{4}{L}, -\frac{6}{L^2}+\frac{12x'}{L^3}, -\frac{6x'}{L^2}+\frac{2}{L}\right] \qquad (4.32)$$

The strain energy follows equation (4.18) and (4.19) to yield the stiffness matrix

$$[K^e] = \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \qquad (4.33)$$

where $a = EI/L^3$.

4.4.2.2 Timoshenko Beam

In Simple beam, the loads are purely bending moment and strain energies are purely from axial tension and compression. To incorporate independent variation for w and γ (shear strain), the strain energy due to shear has to be considered in addition to the strain energy due to direct bending as given by equation (4.28).

$$SE/length(bending) = \frac{1}{2}EI\left(\frac{d\theta}{dx}\right)^2 \qquad (4.34)$$

$$SE/length(shear) = \frac{1}{2}EI(\frac{d\theta}{dx})^2 + \frac{1}{2}\kappa AG\gamma^2 \qquad (4.35)$$

γ is related to the shear force Q by $\gamma = Q/\kappa AG$, the deflection becomes

$$\frac{d\theta}{dx} = \frac{d}{dx}\left(\frac{dw}{dx} - \frac{Q(x)}{\kappa AG}\right) = -\frac{M(x)}{EI} \qquad (4.36)$$

The total strain energy is obtained by integrating over the entire length

$$U^e = \int_0^L \frac{1}{2}\left[EI\left(\frac{d^2w}{dx^2} - \frac{d\gamma}{dx}\right)^2 + \kappa AG\gamma^2\right]dx \qquad (4.37)$$

because $\theta = \mu - \gamma$.

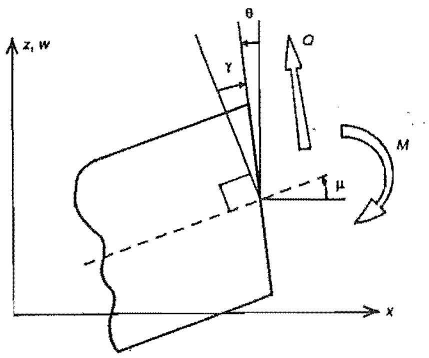


figure 4.3 Shear deformation

Shape functions are now specified for w and γ using cubic interpolation which gives

$$W(x')=n_1(x')w_1+n_2(x')\ \mu+n_3(x')w_2+n_4(x')\ \mu_2 \qquad (4.38)$$

When n_i is defined in equation (4.27).

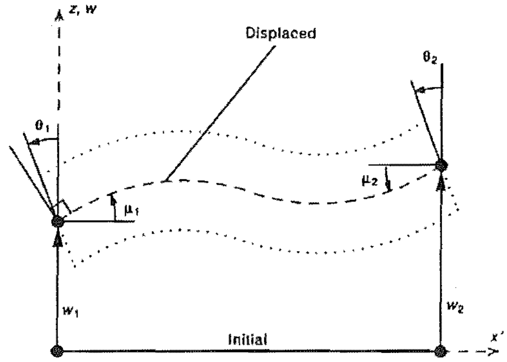


Figure 4.4 Timonshenko beam element

The shear function uses a zeroth order interpolation because the shear stress is assumed constant within tracheid walls.

$$w(x')=n_1(x')w_1+n_2(x')\ (\theta_1+\gamma_o)+n_3(x')w_2+n_4(x')\ (\theta_1+\gamma_o) \qquad (4.39)$$

Rearranging in matrix shape relationship,

$$w(x') = [n_1(x'), n_2(x'), n_3(x'), n_4(x'), n_5(x')]\begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \\ \gamma_o \end{bmatrix} = [N_\gamma^e][d_\gamma^e] \quad (4.40)$$

where $n_5(x')=n_2(x')+n_4(x')$.

In matrix form, equation (4.37) yields

$$U^e = \frac{1}{2}[d_\gamma^e]^T[K_\gamma^e][d_\gamma^e] \qquad (4.41)$$

where the stiffness matrix $[K_\gamma^e]$ has components

$$[K_\gamma^e]=\begin{bmatrix} 12a & 6aL & -12a & 6aL & 12aL \\ 6aL & 4aL^2 & -6aL & 2aL^2 & 6aL^2 \\ -12a & -6aL & 12a & -6aL & -12aL \\ 6aL & 2aL^2 & -6aL & 4aL^2 & 6aL^2 \\ 12aL & 6aL^2 & -12aL & 6aL^2 & (12aL^2+b) \end{bmatrix} \tag{4.42}$$

$a= EI/L^2$ and $b=\kappa AGL$.

To eliminate γ_o, if the loads are applied at the nodal points, the γ_o equation in the assembled form will be equal to zero which simplifies to

$$\gamma_o = \frac{1}{12aL^2+b}[-(12aL)w_1-(6aL^2)\theta_1+(12aL)w_2-(6aL^2)\theta_2] \tag{4.43}$$

Incorporating equation (4.34) into a transformation without γ_o

$$\begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta_2 \\ \gamma_0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -2c & -cL & 2c & -cL \end{bmatrix} \begin{bmatrix} w_1 \\ \theta_1 \\ w_2 \\ \theta 2 \end{bmatrix} \tag{4.44}$$

where $c=6aL/(12aL^2+b)$, or in matrix notation

$$[d_\gamma^e]=[T][d^e] \tag{4.45}$$

Equation (4.41) becomes

$$U^e = \frac{1}{2}[d^e]^T[K^e][d^e] \tag{4.46}$$

where $[K^e]=[T^T][K_\gamma^e][T]$ with components after evaluating the matrix as

$$[K^e]=\left[\frac{a}{1+\varepsilon}\right]\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & (4+\varepsilon)L^2 & -6L & (2-\varepsilon)L^2 \\ -12 & -6L & 12 & -6L \\ 6L & (2-\varepsilon)L^2 & -6L & (4+\varepsilon)L^2 \end{bmatrix} \tag{4.47}$$

where $\varepsilon = \dfrac{12EI}{\kappa AGL^2}$.

4.4.3 Assembly of Global stiffiness matrix

The element stiffness matrix is the finite element building blocks of the continuum problem estimated by discrete elements. Equation (4.23) relates the system's total energy, neglecting the thermal component.

$$\chi = \Sigma^{m}_{e=1}\chi^{e} = 1/2\,[d]^{T}\,[K]\,[d] - [d]^{T}\,[f_g] \qquad (4.48)$$

The term $[K]$ and $[f_g]$ are obtained by incrementing them one element at a time, adding contributions of each element depending on the connectivity of the elements. The procedure to perform the assembly is as follows:

1. Initialise $[K]$ and $[f_g]$ components to zero.
2. Select an element.
3. Calculate $[k^{e}]$ and $[f_g{}^{e}]$ using equation (4.19) and (4.20)and noting the degree of freedom numbers of each of the element $j_1, j_2 \ldots j_n$.
4. Add components of the elemental matrix $k^{e}_{\alpha\beta}$ to global components $k_{j\alpha j\beta}$ for all indices α=1..n, β=1..n.
5. Add components of elemental matrix $(f_g^{e})_\alpha$ to global components $(f_g)_{j\alpha}$ for all indices α=1..n.
6. Return to step 2 and select another element till all elements have been processed, the assembly process is completed now.

4.4.4 Energy due to Concentrated force

The final component of the external concentrated applied load P to the system is now added to the equation (4.48). The concentrated load P acting in the direction of a nodal displacement δ_i would contribute a $-\delta_i P$ to the energy of the system. The final energy equation will be

$$\chi = \Sigma^{m}_{e=1}\chi^{e} - [d]^{T}[f_P] = \frac{1}{2}[d]^{T}[K][d] - [d]^{T}[f] \qquad (4.49)$$

where f is the nodal force vector given by $f = f_g + f_p$, again neglecting the thermal stress effect.

4.4.5 Minimisation and solution of Rayleigh-Ritz method

The energy equation (4.49) is now ready for minimisation by differentiation. The derivatives of the total energy with respect to each displacement must be equal to zero.

$$d\chi = \begin{bmatrix} \partial\chi / \partial\delta_1 \\ \partial\chi / \partial\delta_2 \\ . \\ . \\ \partial\chi / \partial\delta_n \end{bmatrix} = [K][d]-[f]=0 \qquad (4.50)$$

4.4.6 Constraints

The Rayleigh-Ritz solution requires that the partial derivatives of χ are zero for δ_i which are free to move. In order to distinguish easily between constrained and non-constrained displacements, the convention that unconstrained displacements are numbered first and constrained ones last is adopted. In the general case where there are r unconstrained degrees of freedom, $\delta_1 \ldots \delta_r$, and the remainder $\delta_{r+i} \ldots \delta_n$, are constrained. The nature of the constraint varies from problem to problem, the simplest is zero displacement in a particular direction commonly at a point of support or point of symmetry. If such is the case, the column and row of the zero displacement δ_1 is deleted. If two displacements δ1 and δ2 are constrained to move together proportionally by a k such that δ1=kδ2, then the row and column of δ2 displacement are multiplied by the factor k and added to the corresponding row and column of δ1 displacement and the row and column of δ2 displacement are deleted. This system of partitioning and combination of coupled displacements reduce the overall number of degree of freedom and the complexity of the matrix and therefore the increases the ease of solution.

Some knowledge of FEM is necessary understand the above concepts as a preliminary to the understanding of the analytical solution to the wood modeling later in this thesis.

Chapter 5

Analytical solution of Regular hexagon

The earlier part of the work presented in this thesis relates to the numerical simulation or modelling of microscopic wood cell structure using commercial Finite Element codes ANSYS 52. This section will model the wood cell tracheids as cellular honeycomb structure and solve them analytically using the Energy method to compare the results with ANSYS results. This will validate the FE model and provide a better understanding of the behaviour of the model from an analytical point as the analytical models are evolved and refined with increased complexity to bring both the FE model and the analytical model to a closer agreement. Earlier analytical research by Gibson and Ashby [1] on cellular solids provides the background for comparison of the derived Energy solution.

5.1 Gibson's simple equation using beams

5.1.1 Normal Elastic moduli

The cellular structure of wood is shown, idealised as in figure 5.1. The tracheids, which make up the bulk of the cells, can be idealised as regular array of hexagonal prisms with occasional transverse membranes. They are traversed by rays, radial bands of shorter, more rectangular cells and circular sap channels running up the axis of the tree.

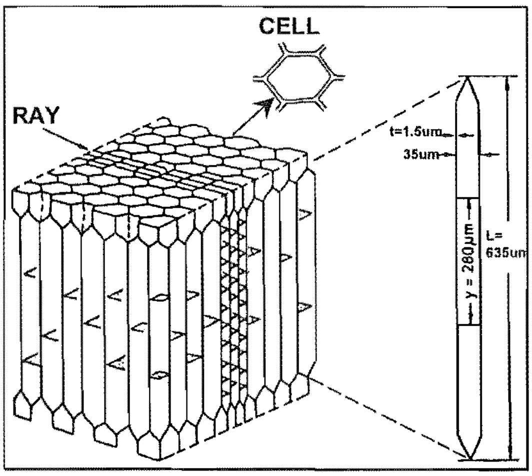


Figure 5.1 Model of idealised wood

By idealising the structure of a typical cell as a hexagonal honeycomb structure. Gibson and Ashby's work in mechanics of honeycomb especially hexagonal honeycomb is the basis of the mechanical model.

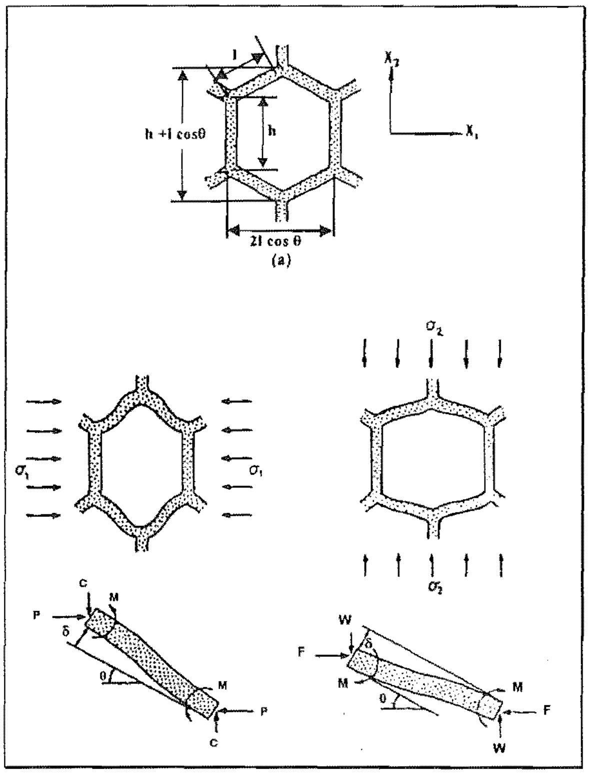


Figure 5.2 Idealised hexagonal wood cell model

Using free-body diagram and standard beam theory, a stress σ_1 in the *x1* direction yield a moment

$$M = Pl/2 \sin \theta \tag{5.1}$$

where $P = \sigma_r (h + l \sin \theta)\, b$.

Substitute the standard beam deflection [52](neglecting axial strain), the wall deflection is

$$\delta = \frac{Pl^3 \sin\theta}{12EI} \tag{5.2}$$

where I is the second moment of inertia of the cell wall $(I = \frac{bt^3}{12}$ for a uniform wall thickness of t) and E is the isotropic Young's modulus for the cell wall material. The strain component in the *x1* direction is given by;

$$\varepsilon_1 = \frac{\delta \sin\theta}{l\cos\theta} = \frac{\sigma_1 (h + l\sin\theta) b l^2 \sin^2\theta}{12EI\cos\theta} \tag{5.3}$$

The Young's modulus of the honeycomb in $x1$-r direction is given by;

$$E_r = \sigma_1 / \varepsilon_1$$

$$= \frac{Et^3 \cos\theta}{l^2 \sin^2\theta (h + l\sin\theta)} \tag{5.4}$$

Loading in the $x2$- t direction gives the force $W=\sigma_t\, lb\, cos\ \theta$ and moment $M=Wl/2\cos\theta$ and the deflection of the wall in the t direction is

$$\varepsilon_2 = \frac{\delta\cos\theta}{h + l\sin\theta} = \frac{\sigma_2 b l^4 \cos^3\theta}{12EI(h + l\sin\theta)} \tag{5.5}$$

from which the Young's modulus in the t direction is derived as;

$$E_t = \frac{Et^3 (h + l\sin\theta)}{l^4 \cos^3\theta} \tag{5.6}$$

The Poisson's ratio is calculated by the negative ratio of the strains normal to and parallel to the loading direction.

$$\nu_{rt} = -\frac{\varepsilon_t}{\varepsilon_r} = \frac{l\cos^2\theta}{(h + l\sin\theta)\sin\theta} \tag{5.7}$$

5.1.2 Shear Modulus

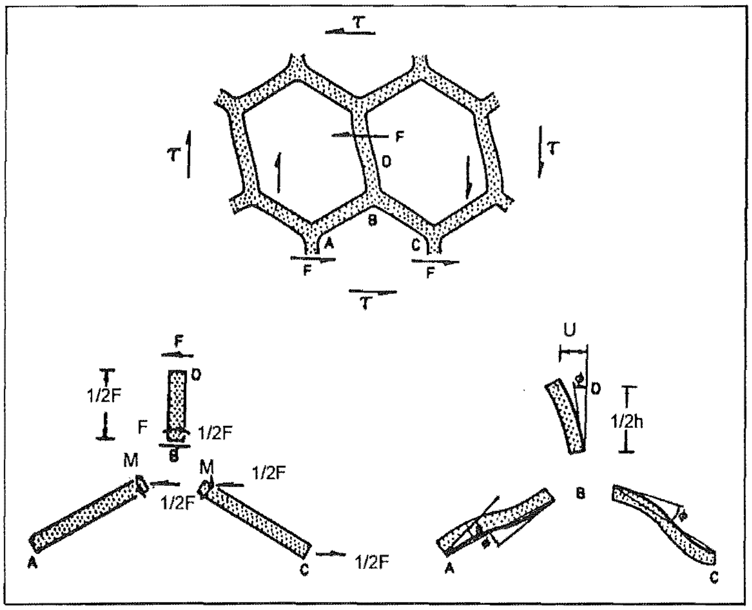


Figure 5.3 Cell deformation by Shear stress

The shear modulus is calculated as shown in the figure 5.3. Because of symmetry, there is no relative motion of points A, B and C when the honeycomb is sheared; the shear deflection U is entirely due to the bending of beam BD and its rotation (ϕ) about the point B. Summing moments at B gives the moment applied to members AB and BC;

$$M = \frac{Fh}{4} \tag{5.8}$$

From standard beam deflection, $\delta = ML^2 / 6EI$, the angle of rotation is

$$\phi = \frac{Fhl}{24EI} \tag{5.9}$$

The deflection U of point D with respect to B is

$$U = \frac{1}{2}\phi h + \frac{F}{3EI}\left(\frac{h}{2}\right)^3 = \frac{Fh^2(I + 2h)}{48EI} \tag{5.10}$$

The shear strain, γ is given by;

$$\gamma = \frac{2U}{(h + l\sin\theta)} = \frac{Fh^2(l + 2h)}{24EI(h + l\sin\theta)} \tag{5.11}$$

The shear stress, $\tau = F / 2lb\cos\theta$, giving the shear modulus

$$G_{rt} = \frac{\tau}{\gamma} = \frac{Et^3(h + l\sin\theta)}{lh^2\cos\theta(l + 2h)} \tag{5.12}$$

These results have been extensively checked by experiments on elastomeric and metal honeycombs [1].

5.2 Gibson's Advanced Equation for honeycomb

In her later work, Gibson [53] derived an advanced equation for elastic moduli considering the effects of both membrane strains and shear deformation of cell walls. The consideration of shear effect, Poisson's effect and the slenderness of the beam element bring the derivation much closer in agreement to the Regular FE model [2].

Using Timoshenko's plate equation [54], the elastic moduli (eqaution 5.4, 5.6 and 5.12) are modified by Timoshenko's beam deflection equation.

$$E_r = \frac{E_s}{\left(1 + (2.4 + 1.5\nu + \cot^2\theta)(t/l)^2\right)}$$

$$= \frac{Et^3\cos\theta}{l^2\sin^2\theta(h + l\sin\theta)\left[1 + \left(2.4 + 1.5\nu + \cot^2\theta\right)\left(\frac{t}{l}\right)^2\right]} \tag{5.13}$$

where E_s is equation (5.4) and ν is the Poisson's ration for the wall material (assume isotropic). Similarly, E_t and G_{rt} are modified by the effect of shear and membrane strains.

$$E_t = \frac{Et^3(h + l\sin\theta)}{l^4\cos^3\theta\left[1 + \left(2.4 + 1.5\nu + \tan^2\theta + \frac{2h}{l\cos^2\theta}\right)\left(\frac{t}{l}\right)^2\right]} \tag{5.14}$$

$$G_{rt} = \frac{Et^3(h+l\sin\theta)}{lh^2\cos\theta(l+2h)\left[1+\left(\frac{(l+h\sin\theta)^2}{2l\cos^2\theta}+\frac{6h(2l+h)(1+\nu)}{5l}\right)\left(\frac{2t^2}{h^2(2h+l)}\right)\right]} \quad (5.15)$$

5.3 Derivation of Gibson's simple equation using FE method

Gibson in her research on cellular materials has used simple beam to predict the elastic moduli and later an advanced version taking effect into consideration axial tension and membrane shear. Another approach using energy formulation and stiffness matrix is evaluated.

This particular approach is advantageous in model development as further refinements can build upon earlier model using increasingly complex considerations and basically the same methodology of solution. The stiffness method of course is closely linked to the FE method.

To understand the methodology, the energy method will be use to duplicate Gibson's derivation of the simple equation. This model will validate the FE method against a known solution. It will also serve to explain the procedure in manipulating and solving the model.

5.3.1 Elastic Modulus

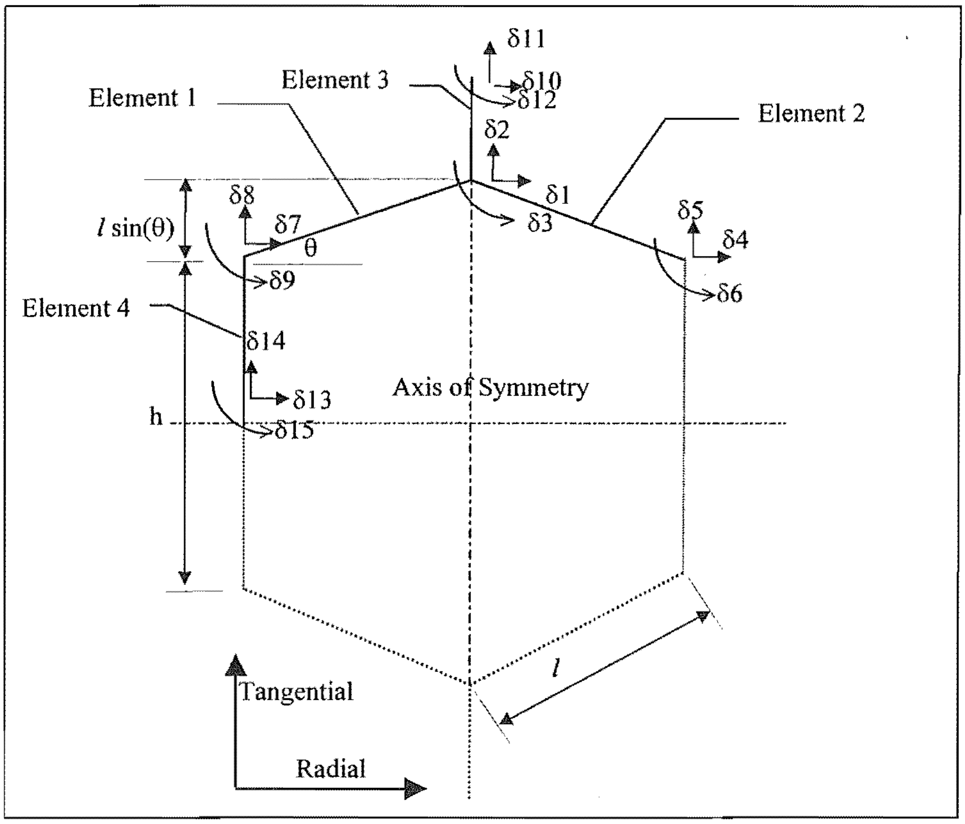


Figure 5.4 Wood cell modelled as four FE elements

By symmetry, the wood cell can be modelled as four beam elements subjected to specific boundary constraints. Firstly, we will assembly the global stiffness matrix by defining the individual stiffness matrix of each element and following the procedure listed in section 4.4.3.

The simple beam element stiffness matrix is as derived in section 4.4.2.1 when expanded into three degrees of freedom within its local co-ordinates system is:

$$\begin{bmatrix} K & 0 & 0 & -K & 0 & 0 \\ 0 & 12a & 6aL & 0 & -12a & 6aL \\ 0 & 6aL & 4aL^2 & 0 & -6aL & 2aL^2 \\ -K & 0 & 0 & K & 0 & 0 \\ 0 & -12a & -6aL & 0 & 12a & -6aL \\ 0 & 6aL & 2aL^2 & 0 & -6aL & 4aL^2 \end{bmatrix} \quad (5.20)$$

where $a = EI/L^3$, $I = b\,t^3/12$, $k = EA/L$, L being the element length.

To transform the elemental stiffness matrix to its global co-ordinates system, the transformation matrix T is used;

$$[K^e]=[T^e]^T[K^{e'}][T^e] \tag{5.21}$$

where $[K^e]$ is the elemental stiffness matrix in global co-ordinates, $[T^e]$ is the transformation matrix given by equation (5.22) and $[K^{e'}]$ is the stiffness matrix in the local co-ordinates as in equation (5.20).

The transformation matrix, $[T^e]$ is given by:

$$\begin{bmatrix} -\cos(\theta) & \sin(\theta) & 0 & 0 & 0 & 0 \\ -\sin(\theta) & -\cos(\theta) & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\cos(\theta) & \sin(\theta) & 0 \\ 0 & 0 & 0 & -\sin(\theta) & -\cos(\theta) & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.22}$$

Hence the general element global matrix in equation (5.21) becomes:

$$
\begin{bmatrix}
\frac{kl^3\cos^2\theta + Ebt^3\sin^2\theta}{l^3} & -\frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & -\frac{Ebt^3\sin\theta}{2l^2} & -\frac{kl^3\cos^2\theta + Ebt^3\sin^2\theta}{l^3} & \frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & -\frac{Ebt^3\sin\theta}{2l^2} \\
-\frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & \frac{kl^3\sin^2\theta + Ebt^3\cos^2\theta}{l^3} & \frac{Ebt^3\cos\theta}{2l^2} & \frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & -\frac{kl^3\sin^2\theta + Ebt^3\cos^2\theta}{l^3} & \frac{Ebt^3\cos\theta}{2l^2} \\
-\frac{Ebt^3\sin\theta}{2l^2} & \frac{Ebt^3\cos\theta}{2l^2} & \frac{Ebt^3}{3l} & \frac{Ebt^3\sin\theta}{2l^2} & -\frac{Ebt^3\cos\theta}{2l^2} & \frac{Ebt^3}{6l} \\
-\frac{kl^3\cos^2\theta + Ebt^3\sin^2\theta}{l^3} & \frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & \frac{Ebt^3\sin\theta}{2l^2} & \frac{kl^3\cos^2\theta + Ebt^3\sin^2\theta}{l^3} & -\frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & \frac{Ebt^3\sin\theta}{2l^2} \\
\frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & -\frac{kl^3\sin^2\theta + Ebt^3\cos^2\theta}{l^3} & -\frac{Ebt^3\cos\theta}{2l^2} & -\frac{\cos\theta\sin\theta\left(-kl^3 + Ebt^3\right)}{l^3} & \frac{kl^3\sin^2\theta + Ebt^3\cos^2\theta}{l^3} & -\frac{Ebt^3\cos\theta}{2l^2} \\
-\frac{Ebt^3\sin\theta}{2l^2} & \frac{Ebt^3\cos\theta}{2l^2} & \frac{Ebt^3}{6l} & \frac{Ebt^3\sin\theta}{2l^2} & -\frac{Ebt^3\cos\theta}{2l^2} & \frac{Ebt^3}{3l}
\end{bmatrix}
$$

equation (5.23)

Substitution element 1stiffness matrix into equation (5.21), global element stiffness matrix [$k1$] is given by

$$k1 := \begin{bmatrix} \%1 & -\%2 & \%3 & -\%1 & \%2 & \%3 \\ -\%2 & \%4 & \%5 & \%2 & -\%4 & \%5 \\ \%3 & \%5 & \frac{1}{3}\frac{E\,b\,t^3}{l} & \%6 & \%7 & \frac{1}{6}\frac{E\,b\,t^3}{l} \\ -\%1 & \%2 & \%6 & \%1 & -\%2 & \%6 \\ \%2 & -\%4 & \%7 & -\%2 & \%4 & \%7 \\ \%3 & \%5 & \frac{1}{6}\frac{E\,b\,t^3}{l} & \%6 & \%7 & \frac{1}{3}\frac{E\,b\,t^3}{l} \end{bmatrix}$$

$$\%1 := \frac{E\,b\,t\left(\cos(\theta)^2\,l^2 + \sin(\theta)^2\,t^2\right)}{l^3}$$

$$\%2 := \frac{\cos(\theta)\,E\,b\,t\,\sin(\theta)\left(-l^2 + t^2\right)}{l^3}$$

$$\%3 := -\frac{1}{2}\frac{\sin(\theta)\,E\,b\,t^3}{l^2}$$

$$\%4 := \frac{E\,b\,t\left(\sin(\theta)^2\,l^2 + \cos(\theta)^2\,t^2\right)}{l^3} \qquad (5.24)$$

$$\%5 := \frac{1}{2}\frac{\cos(\theta)\,E\,b\,t^3}{l^2}$$

$$\%6 := \frac{1}{2}\frac{\sin(\theta)\,E\,b\,t^3}{l^2}$$

$$\%7 := -\frac{1}{2}\frac{\cos(\theta)\,E\,b\,t^3}{l^2}$$

Element 2 global element stiffness matrix, [$k2$], is similar to element 1 but the angle is $-\theta$, global stiffness matrix [$k3$] and [$k4$] are equal since θ is 90° for both, therefore sin θ=1 and cos θ=0.

[$K3$] and [$k4$] is given by:

$$\begin{bmatrix} 8\frac{E\,b\,t^3}{h^3} & 0 & -2\frac{E\,b\,t^3}{h^2} & -8\frac{E\,b\,t^3}{h^3} & 0 & -2\frac{E\,b\,t^3}{h^2} \\ 0 & 2\frac{E\,b\,t}{h} & 0 & 0 & -2\frac{E\,b\,t}{h} & 0 \\ -2\frac{E\,b\,t^3}{h^2} & 0 & \frac{2}{3}\frac{E\,b\,t^3}{h} & 2\frac{E\,b\,t^3}{h^2} & 0 & \frac{1}{3}\frac{E\,b\,t^3}{h} \\ -8\frac{E\,b\,t^3}{h^3} & 0 & 2\frac{E\,b\,t^3}{h^2} & 8\frac{E\,b\,t^3}{h^3} & 0 & 2\frac{E\,b\,t^3}{h^2} \\ 0 & -2\frac{E\,b\,t}{h} & 0 & 0 & 2\frac{E\,b\,t}{h} & 0 \\ -2\frac{E\,b\,t^3}{h^2} & 0 & \frac{1}{3}\frac{E\,b\,t^3}{h} & 2\frac{E\,b\,t^3}{h^2} & 0 & \frac{2}{3}\frac{E\,b\,t^3}{h} \end{bmatrix} \qquad (5.25)$$

The process of assembly of the Global stiffness matrix begins with element 1, element 2, element 3 and 4, in accordance to the procedure detailed in section 4.4.3, resulting in a massive 15 by 15 global stiffness matrix shown on the next page.

$$
\begin{bmatrix}
8\frac{Ebt^3}{h^3}+2\%1 & 0 & -2\frac{Ebt^3}{h^2}+\frac{\sin(\theta)Ebt^3}{l^2} & -\%1 & -\%2 & \%3 & -\%1 & \%2 & \%3 & -8\frac{Ebt^3}{h^3} & 0 & -2\frac{Ebt^3}{h^2} & 0 & 0 & 0 \\
0 & 2\frac{Ebt}{h}+2\%4 & 0 & -\%2 & -\%4 & \%5 & \%2 & -\%4 & \%6 & 0 & -2\frac{Ebt}{h} & 0 & 0 & 0 & 0 \\
-2\frac{Ebt^3}{h^2}+\frac{\sin(\theta)Ebt^3}{l^2} & 0 & \frac{2}{3}\frac{Ebt^3}{h}+\frac{2}{3}\frac{Ebt^3}{l} & \%7 & \%6 & \frac{1}{6}\frac{Ebt^3}{l} & \%7 & \%5 & \frac{1}{6}\frac{Ebt^3}{l} & 2\frac{Ebt^3}{h^2} & 0 & \frac{1}{3}\frac{Ebt^3}{h} & 0 & 0 & 0 \\
-\%1 & -\%2 & \%7 & \%1 & \%2 & \%7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\%2 & -\%4 & \%6 & \%2 & \%4 & \%6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\%3 & \%5 & \frac{1}{6}\frac{Ebt^3}{l} & \%7 & \%6 & \frac{1}{3}\frac{Ebt^3}{l} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-\%1 & \%2 & \%7 & 0 & 0 & 0 & 8\frac{Ebt^3}{h^3}+\%1 & -\%2 & 2\frac{Ebt^3}{h^2}-\frac{1}{2}\frac{\sin(\theta)Ebt^3}{l^2} & 0 & 0 & 0 & -8\frac{Ebt^3}{h^3} & 0 & 2\frac{Ebt^3}{h^2} \\
\%2 & -\%4 & \%5 & 0 & 0 & 0 & -\%2 & 2\frac{Ebt}{h}+\%4 & \%5 & 0 & 0 & 0 & 0 & -2\frac{Ebt}{h} & 0 \\
\%3 & \%6 & \frac{1}{6}\frac{Ebt^3}{l} & 0 & 0 & 0 & 2\frac{Ebt^3}{h^2}-\frac{1}{2}\frac{\sin(\theta)Ebt^3}{l^2} & \%5 & \frac{2}{3}\frac{Ebt^3}{h}+\frac{1}{3}\frac{Ebt^3}{l} & 0 & 0 & 0 & -2\frac{Ebt^3}{h^2} & 0 & \frac{1}{3}\frac{Ebt^3}{h} \\
-8\frac{Ebt^3}{h^3} & 0 & 2\frac{Ebt^3}{h^2} & 0 & 0 & 0 & 0 & 0 & 0 & 8\frac{Ebt^3}{h^3} & 0 & 2\frac{Ebt^3}{h^2} & 0 & 0 & 0 \\
0 & -2\frac{Ebt}{h} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2\frac{Ebt}{h} & 0 & 0 & 0 & 0 \\
-2\frac{Ebt^3}{h^2} & 0 & \frac{1}{3}\frac{Ebt^3}{h} & 0 & 0 & 0 & 0 & 0 & 0 & 2\frac{Ebt^3}{h^2} & 0 & \frac{2}{3}\frac{Ebt^3}{h} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -8\frac{Ebt^3}{h^3} & 0 & -2\frac{Ebt^3}{h^2} & 0 & 0 & 0 & 8\frac{Ebt^3}{h^3} & 0 & -2\frac{Ebt^3}{h^2} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & -2\frac{Ebt}{h} & 0 & 0 & 0 & 0 & 0 & 2\frac{Ebt}{h} & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2\frac{Ebt^3}{h^2} & 0 & \frac{1}{3}\frac{Ebt^3}{h} & 0 & 0 & 0 & -2\frac{Ebt^3}{h^2} & 0 & \frac{2}{3}\frac{Ebt^3}{h}
\end{bmatrix}
$$

$$\%1 := \frac{Ebt\left(\cos(\theta)^2 l^2 + \sin(\theta)^2 t^2\right)}{l^3}$$

$$\%2 := \frac{\cos(\theta)Ebt\sin(\theta)\left(-l^2+t^2\right)}{l^3}$$

$$\%3 := \frac{1}{2}\frac{\sin(\theta)Ebt^3}{l^2}$$

$$\%4 := \frac{Ebt\left(\sin(\theta)^2 l^2 + \cos(\theta)^2 t^2\right)}{l^3}$$

$$\%5 := \frac{1}{2}\frac{\cos(\theta)Ebt^3}{l^2}$$

$$\%6 := -\frac{1}{2}\frac{\cos(\theta)Ebt^3}{l^2}$$

$$\%7 := -\frac{1}{2}\frac{\sin(\theta)Ebt^3}{l^2}$$

Equation (5.26) Global stiffness matrix

In order to reduce the global matrix further, we will now apply the boundary conditions.

5.3.1.2 Boundary constraints

The constraints for both E_r and E_t are the same and given by:

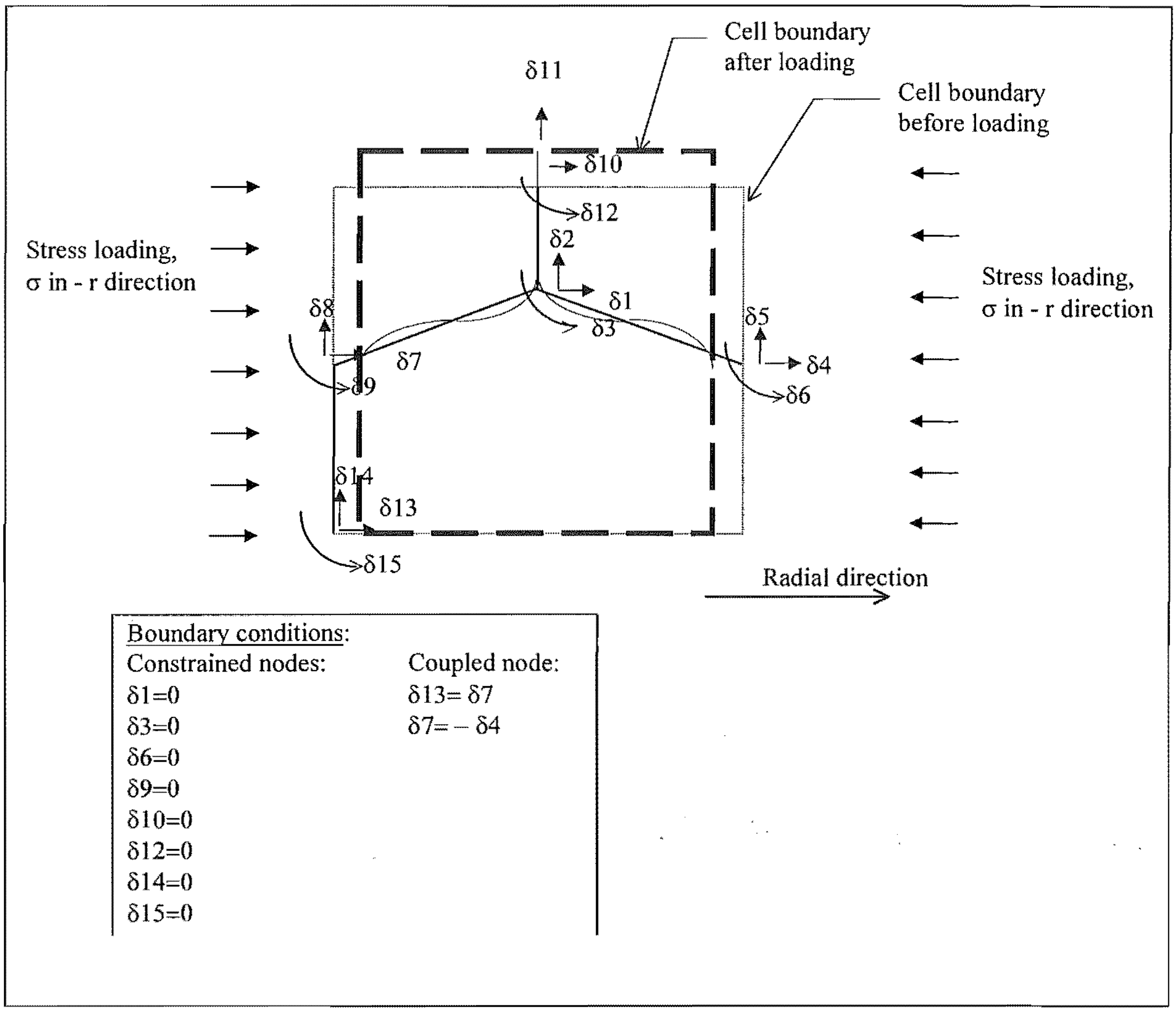


Figure 5.5 Boundary constraints for elastic modulus E_r and E_t only.

The list of constrained degree of freedoms are due to symmetry and the coupled nodes are due to the cyclic constrains. By the process described in 4.4.6, any constrained zero degree of freedom's column and row are deleted and any coupled degree of freedom's column and row are added to the column and row of the constrained degree of freedom and the earlier degree of freedom's column and row are then deleted. For example, row 1 and column 1 of the global stiffness matrix are deleted because the row and column defining $\delta 1$ are constrained as zero. For constrain $\delta 7=-\delta 4$, the negative values of row and column of $\delta 4$ are added to the corresponding row and column values of $\delta 7$ and finally, the row and column of $\delta 4$ are deleted.

Imposing the boundary conditions, and deleting rows and columns where δ equal to zero and coupling δ7, δ4 and the global stiffness matrix, [K], is reduced to a 4 x 4 matrix for displacement δ2, δ7, δ8 and δ11. These degree of freedom define the rest of the displacement of the cell.

$$\begin{bmatrix} 2\frac{E\,b\,t}{h}+2\,\%1 & -2\,\%2 & -2\,\%1 & -2\frac{E\,b\,t}{h} \\ -2\,\%2 & 2\frac{E\,b\,t\left(\cos(\theta)^2 l^2+\sin(\theta)^2 t^2\right)}{l^3} & 2\,\%2 & 0 \\ -2\,\%1 & 2\,\%2 & 2\frac{E\,b\,t}{h}+2\,\%1 & 0 \\ -2\frac{E\,b\,t}{h} & 0 & 0 & 2\frac{E\,b\,t}{h} \end{bmatrix}$$

$$\%1 := \frac{E\,b\,t\left(\sin(\theta)^2 l^2+\cos(\theta)^2 t^2\right)}{l^3} \qquad (5.27)$$

$$\%2 := \frac{\cos(\theta)\,E\,b\,t\,\sin(\theta)\left(l^2-t^2\right)}{l^3}$$

Equation 5.27 is then substituted into equation 4.49.

$$\chi = 2\,(1/2\,[d^T\,][K]\,[d]) - 2([d^T][f][d]) \qquad (5.28)$$

where $[d]=\begin{bmatrix}\delta 2\\ \delta 7\\ \delta 8\\ \delta 11\end{bmatrix}$ and $[f]=\begin{bmatrix}0\\ 2\sigma(h+l\sin(\theta))\\ 0\\ 0\end{bmatrix}$ and [K] is given by equation (5.27)

The equation (5.28) is differentiated and the derivatives are equated to zero.

$$d\chi = \begin{bmatrix} \frac{\partial \chi}{\partial \delta 2} \\ \frac{\partial \chi}{\partial \delta 7} \\ \frac{\partial \chi}{\partial \delta 8} \\ \frac{\partial \chi}{\partial \delta 11} \end{bmatrix} = 0 \qquad (5.29)$$

The four derivative equations are then solved for each displacement;

$\partial\chi / \partial\delta 2 =$

$$2\,E\,b\,t\left(2\,\delta 2\,l^3 + 2\,\delta 2\,\sin(\theta)^2\,h\,l^2 + 2\,\delta 2\,\cos(\theta)^2\,t^2\,h - 2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2 + 2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2 \right.$$
$$\left. - 2\,\delta 8\,h\,\sin(\theta)^2\,l^2 - 2\,\delta 8\,h\,\cos(\theta)^2\,t^2 - 2\,\delta 11\,l^3\right)\Big/\left(h\,l^3\right)$$

(5.30)

$\partial\chi / \partial\delta 7 =$

$$2\,E\,b\,t\left(-2\,\delta 2\,\cos(\theta)\,\sin(\theta)\,h\,l^2 + 2\,\delta 2\,\cos(\theta)\,\sin(\theta)\,h\,t^2 + 2\,\delta 7\,h\,\cos(\theta)^2\,l^2 + 2\,\delta 7\,h\,\sin(\theta)^2\,t^2 \right.$$
$$\left. + 2\,h\,\delta 8\,\cos(\theta)\,\sin(\theta)\,l^2 - 2\,h\,\delta 8\,\cos(\theta)\,\sin(\theta)\,t^2\right)\Big/\left(h\,l^3\right)$$

(5.31)

$\partial\chi / \partial\delta 8 =$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2 - 2\,\delta 2\,\cos(\theta)^2\,t^2\,h + 2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2 - 2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2 + 2\,\delta 8\,l^3 \right.$$
$$\left. + 2\,\delta 8\,h\,\sin(\theta)^2\,l^2 + 2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)\Big/\left(h\,l^3\right)$$

(5.32)

$$\partial\chi / \partial\delta 11 = 2\,\frac{E\,b\,t\left(-2\,\delta 2\,l^3 + 2\,\delta 11\,l^3\right)}{h\,l^3} - 4\,\sigma\,b\,l\,\cos(\theta)$$

(5.33)

The derivatives yield the following results;

$$\delta 2 = \frac{\cos(\theta)\,\sin(\theta)\left(t^2 - l^2\right)l\,\sigma\,(h + l\,\sin(\theta))}{E\,t^3}$$

(5.30A)

$$\delta 7 = -\frac{l\left(\sin(\theta)^2\,l^2 + \cos(\theta)^2\,t^2\right)\sigma\,(h + l\,\sin(\theta))}{E\,t^3}$$

(5.31A)

$\delta 8 = 0$

$$\delta 11 = \frac{\cos(\theta)\sin(\theta)\left(-l^2+t^2\right) l\,\sigma\,(h+l\sin(\theta))}{E\,t^3} \tag{5.32A}$$

The strain of the model in the radial direction is given by

$$\varepsilon_r = \delta 7 / L \cos(\theta)$$

$$= -\frac{\left(\sin(\theta)^2 l^2 + \cos(\theta)^2 t^2\right)\sigma\,(h+l\sin(\theta))}{E\,t^3\cos(\theta)} \tag{5.33}$$

The elastic modulus in the radial direction is then

$E_r = stress/strain$

$$= -\frac{E\,t^3\cos(\theta)}{\left(\sin(\theta)^2 l^2 + \cos(\theta)^2 t^2\right)(h+l\sin(\theta))}$$

$$= \frac{E\,t^3\cos(\theta)}{l^2\sin(\theta)^2\left(1+\frac{t^2\cot(\theta)^2}{l^2}\right)(h+l\sin(\theta))} \tag{5.34}$$

Neglecting the effect of axial loading terms; $(t/l)^2 \cot^2\theta$,

$$E_r = \frac{E\,t^3\cos(\theta)}{l^2\sin(\theta)^2(h+l\sin(\theta))}$$ (Gibson's simplified equation) (5.35)

The strain in the tangential due to strain in the radial direction is given by:

Strain $\varepsilon_{rt} = \delta 11 / (h + L\sin(\theta))$

$$= \frac{\cos(\theta)\sin(\theta)\left(-l^2+t^2\right) l\,\sigma}{E\,t^3} \tag{5.36}$$

Poison ratio of the model is given by,

$$\nu_{rt} = -\varepsilon_{rt}/\varepsilon_r$$

$$= -\frac{\cos(\theta)^2 \sin(\theta) \left(-l^2 + t^2\right) l}{\left(\sin(\theta)^2 l^2 + \cos(\theta)^2 t^2\right)(h + l \sin(\theta))} \qquad (5.37)$$

To compute E_t, we redefine the load on the model.

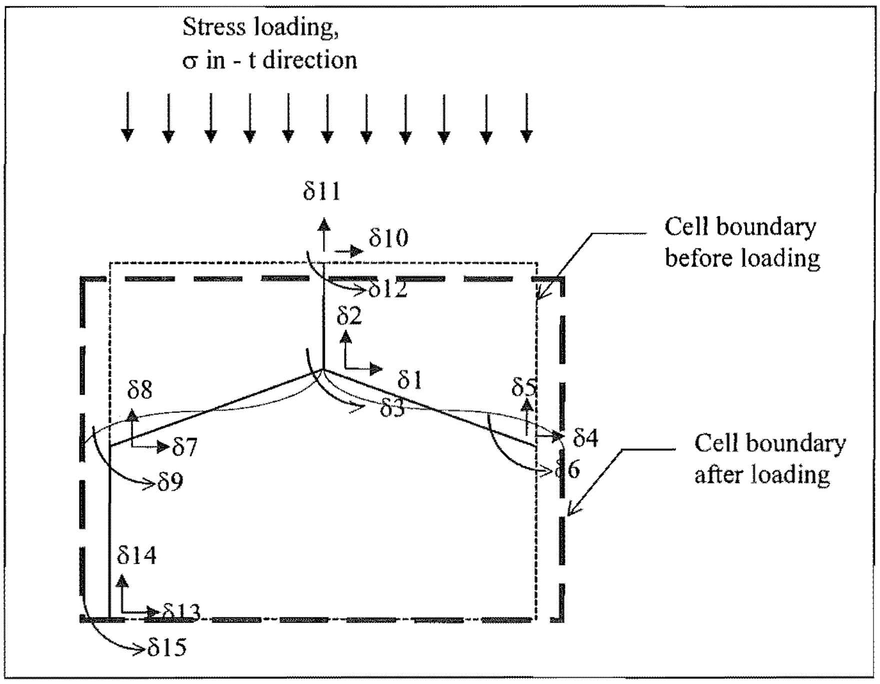


Figure 5.6 Load case for E_t computation.

Following the same procedure as above but with a different force matrix on δ11;

$$[f] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -2\sigma bl\cos(\theta) \end{bmatrix} \qquad (5.38)$$

Formulating the energy equation (5.28) and solving the derivatives (5.29) yields,

$$\delta 2 = \frac{\sigma l \cos(\theta)(\cos(\theta)^2 l^3 + t^2 h + t^2 l - l\cos(\theta)^2 t^2)}{t^3 E} \qquad (5.39)$$

$$\delta 7 = \frac{\left(l^2 \cos(\theta)^2 \sin(\theta)\left(l \;\; - t^2\right)\sigma\right)}{t^3 E} \qquad (5.40)$$

$$\delta 8 = \frac{h\sigma l\cos(\theta)}{Et} \qquad (5.41)$$

$$\delta 11 = \frac{2\sigma l \cos(\theta)\left(\cos(\theta)^2 l^3 + 2t^2 h + t^2 l - l\cos(\theta)^2 t^2\right)}{t^3 E} \tag{5.42}$$

Strain in the tangential direction is equal to;

$\varepsilon_t = \delta 11 / (h + L \sin(\theta))$

$$= \frac{\sigma l \cos(\theta)\left(\cos(\theta)^2 l^3 + 2t^2 h + t^2 l - l\cos(\theta)^2 t^2\right)}{t^3 E(h + l\sin(\theta))} \tag{5.43}$$

$$E_t = \frac{t^3 E\,(h + l\sin(\theta))}{\cos(\theta)\, l\left(2t^2 h + t^2 \sin(\theta)^2 l + \cos(\theta)^2 l^3\right)}$$

$$= \frac{t^3 E\,(h + l\sin(\theta))}{l^4 \cos(\theta)^3 \left(1 + 2\dfrac{t^2 h}{l^3 \cos(\theta)^2} + \dfrac{t^2 \tan(\theta)^2}{l^2}\right)} \tag{5.44}$$

Neglecting the effect of axial loading terms in the denominator;

$$2\frac{t^2 h}{l^3 \cos(\theta)^2} + \frac{t^2 \tan(\theta)^2}{l^2},$$

$$E_t = \frac{t^3 E\,(h + l\sin(\theta))}{l^4 \cos(\theta)^3} \qquad \text{(Gibson's simplified equation)} \tag{5.45}$$

5.3.2 Shear Modulus

The constraints for shear modulus are different from the elastic modulus since the shear deformations and boundary conditions are different from the elastic model, but the basic global stiffness matrix prior to constrain application is similar.

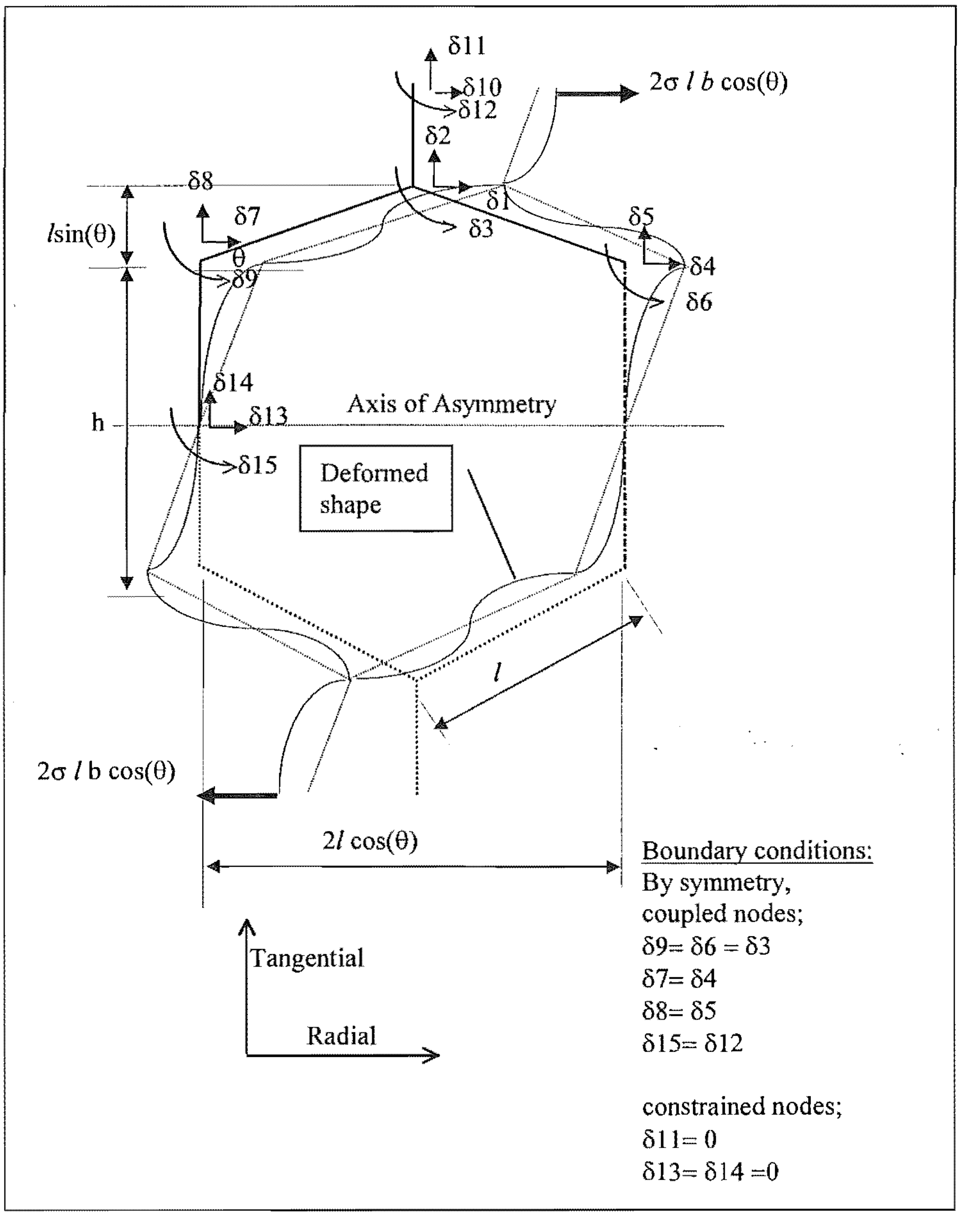


Figure 5.7 Model for shear modulus

By symmetry, we can model half of the cell for shear calculation. By a similar process of assembling the global matrix and imposing the boundary conditions and coupling the relevant nodes (necessary for cyclic tiling), the stiffness matrix is reduced to 7 nodes, δ1, δ2, δ3, δ7, δ8, δ10 and δ12.

$$
\begin{bmatrix}
8\frac{E b t^3}{h^3}+2\%1, & 0, & 2\frac{\sin(\theta) E b t^3}{l^2}-2\frac{E b t^3}{h^2}, & -2\%1, & 0, & -8\frac{E b t^3}{h^3}, & -2\frac{E b t^3}{h^2} \\
0, & 2\frac{E b t}{h}+2\%2, & 0, & 0, & -2\%2, & 0, & 0 \\
2\frac{\sin(\theta) E b t^3}{l^2}-2\frac{E b t^3}{h^2}, & 0, & \frac{4}{3}\frac{E b t^3}{h}+2\frac{E b t^3}{l}, & 2\frac{E b t^3}{h^2}-2\frac{\sin(\theta) E b t^3}{l^2}, & 0, & 2\frac{E b t^3}{h^2}, & \frac{2}{3}\frac{E b t^3}{h} \\
-2\%1, & 0, & 2\frac{E b t^3}{h^2}-2\frac{\sin(\theta) E b t^3}{l^2}, & 8\frac{E b t^3}{h^3}+2\%1, & 0, & 0, & 2\frac{E b t^3}{h^2} \\
0, & -2\%2, & 0, & 0, & 2\frac{E b t}{h}+2\%2, & 0, & 0 \\
-8\frac{E b t^3}{h^3}, & 0, & 2\frac{E b t^3}{h^2}, & 0, & 0, & 8\frac{E b t^3}{h^3}, & 2\frac{E b t^3}{h^2} \\
-2\frac{E b t^3}{h^2}, & 0, & \frac{2}{3}\frac{E b t^3}{h}, & 2\frac{E b t^3}{h^2}, & 0, & 2\frac{E b t^3}{h^2}, & \frac{4}{3}\frac{E b t^3}{h}
\end{bmatrix}
$$

$$\%1 := \frac{E b t \left(\cos(\theta)^2 l^2 + \sin(\theta)^2 t^2\right)}{l^3}$$

$$\%2 := \frac{E b t \left(\sin(\theta)^2 l^2 + \cos(\theta)^2 t^2\right)}{l^3} \qquad (5.46)$$

The load matrix will be given by on node 10;

$$[f] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 2\sigma l \cos(\theta) b \\ 0 \end{bmatrix} \qquad (5.47)$$

Evaluating the energy equation (5.28) and differentiating $\partial\chi/\partial\delta_1$,

$\partial\chi / \partial\delta 1 =$

$$2 E b t\left(-2\,\delta 2 \sin(\theta)^2 h l^2 - 2\,\delta 2 \cos(\theta)^2 t^2 h + 2\,\delta 7 \cos(\theta)\sin(\theta) h l^2 - 2\,\delta 7 \cos(\theta)\sin(\theta) h t^2 + 2\,\delta 8\, l^3 + 2\,\delta 8\, h \sin(\theta)^2 l^2 + 2\,\delta 8\, h \cos(\theta)^2 t^2\right) / \left(h l^3\right)$$

$$(5.48)$$

$\partial\chi / \partial\delta 2 =$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.49)

$\partial\chi\,/\,\partial\delta 3=$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.50)

$\partial\chi\,/\,\partial\delta 7=$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.51)

$\partial\chi\,/\,\partial\delta 8=$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.52)

$\partial\chi\,/\,\partial\delta 10=$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.53)

$\partial\chi\,/\,\partial\delta 12=$

$$2\,E\,b\,t\left(-2\,\delta 2\,\sin(\theta)^2\,h\,l^2-2\,\delta 2\,\cos(\theta)^2\,t^2\,h+2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,l^2-2\,\delta 7\,\cos(\theta)\,\sin(\theta)\,h\,t^2+2\,\delta 8\,l^3\right.$$
$$\left.+2\,\delta 8\,h\,\sin(\theta)^2\,l^2+2\,\delta 8\,h\,\cos(\theta)^2\,t^2\right)/\left(h\,l^3\right)$$

(5.54)

Equating to zero and solving the equations give the respective displacements;

$$\delta 1=\ -\frac{1}{2}\,\frac{\left(h^2\,\sin(\theta)^2\,t^2+3\,\sin(\theta)\,t^2\,h\,l+2\,l^2\,t^2+2\,h^3\,\cos(\theta)^2\,l+\cos(\theta)^2\,l^2\,h^2\right)\sigma}{\cos(\theta)\,t^3\,E} \tag{5.55}$$

$\delta 2=0$

$$\delta 3 = \frac{\left(h \sin(\theta)^2 t^2 + l\, t^2 \sin(\theta) + \cos(\theta)^2 l^2 h\right) \sigma}{\cos(\theta)\, t^3 E} \tag{5.56}$$

$$\delta 7 = -\frac{1}{2} \frac{\sigma h \left(h \sin(\theta)^2 t^2 + l\, t^2 \sin(\theta) + 2 \cos(\theta)^2 l\, h^2 + \cos(\theta)^2 l^2 h\right)}{\cos(\theta)\, t^3 E} \tag{5.57}$$

$$\delta 8 = 0$$

$$\delta 10 = -\frac{\left(h^2 \sin(\theta)^2 t^2 + 2 \sin(\theta)\, t^2 h\, l + 2 h^3 \cos(\theta)^2 l + \cos(\theta)^2 l^2 h^2 + l^2 t^2\right) \sigma}{\cos(\theta)\, t^3 E} \tag{5.58}$$

$$\delta 12 = \frac{\sigma \left(h \sin(\theta)^2 t^2 + l\, t^2 \sin(\theta) + 3 \cos(\theta)^2 l\, h^2 + \cos(\theta)^2 l^2 h\right)}{\cos(\theta)\, t^3 E} \tag{5.59}$$

Shear strain is then the expression, $\gamma = \delta 10 / (h + l \sin(\theta))$;

$$\gamma = -\frac{\sigma \left(\cos(\theta)^2 l^2 h^2 + 2 h^3 \cos(\theta)^2 l + \sin(\theta)^2 t^2 h^2 + 2 \sin(\theta)\, t^2 h\, l + t^2 l^2\right)}{t^3 \cos(\theta)\, E\, (h + l \sin(\theta))} \tag{5.60}$$

Shear Modulus $G_{rt} = \tau / \gamma$

$$G_{rt} = \frac{t^3 E(h + l\sin(\theta))}{h^2 l \cos(\theta)(2h + l)\left(1 + \dfrac{t^2 (l + h\sin(\theta))^2}{h^2 (l + 2h) l \cos(\theta)^2}\right)} \tag{5.61}$$

Neglecting the second term of denominator (axial loading), gives the Gibson and Ashby's simplified equation.

$$G_{rt} = \frac{Et^3 (h + l\sin\theta)}{lh^2 \cos\theta (l + 2h)} \tag{5.62}$$

The above computations are extremely laborious when performed by hand and in this case the equations are best solved by equation manipulation program such as Maple V, only the final expression such as equation (5.35), (5.37), (5.45) and (5.62) are expressed by hand into an appropriate form for presentation. The rest of the FE models presented follow similar method of manipulation.

5.4 Derivation of Gibson's Advanced Equation using Timoshenko's Beam

Gibson's advanced equation is now derived using Timoshenko beam as in section 4.4.2.2. The basic procedure for simple beam is followed but the Timoshenko deep beam stiffness matrix as in equation (4.47) replaces the element stiffness matrix $[K^e]$ in equation (5.20).

The boundary conditions for all load case of E_r, E_t and G_{rt} are the same and so is the force matrix.

The solution of energy equation for E_r gives the following displacements

$$\delta 2 := \frac{\left(\kappa l^2 + 2 t^2 + 2 t^2 \nu - t^2 \kappa\right) \sin(\theta) \cos(\theta) \, l \, \sigma \, (h + l \sin(\theta))}{\left(\sin(\theta)^4 + \cos(\theta)^4 + 2 \cos(\theta)^2 \sin(\theta)^2\right) \kappa t^3 E} \tag{5.63}$$

$$\delta 7 := \frac{\left(\sin(\theta)^2 \kappa l^2 + 2 \sin(\theta)^2 t^2 + 2 \sin(\theta)^2 t^2 \nu + \cos(\theta)^2 t^2 \kappa\right) l \, \sigma \, (h + l \sin(\theta))}{\left(\sin(\theta)^4 + \cos(\theta)^4 + 2 \cos(\theta)^2 \sin(\theta)^2\right) \kappa t^3 E} \tag{5.64}$$

$$\delta 8 := 0$$

$$\delta 11 := \frac{\left(\kappa l^2 + 2 t^2 + 2 t^2 \nu - t^2 \kappa\right) \sin(\theta) \cos(\theta) \, l \, \sigma \, (h + l \sin(\theta))}{\left(\cos(\theta)^4 + \sin(\theta)^4 + 2 \cos(\theta)^2 \sin(\theta)^2\right) \kappa t^3 E} \tag{5.65}$$

Strain in the radial direction ε_r is then

$$\varepsilon_r = \frac{\delta 7}{l \cos \theta}$$

$$= \frac{\left(\sin(\theta)^2 \kappa l^2 + 2 \sin(\theta)^2 t^2 + 2 \sin(\theta)^2 t^2 \nu + \cos(\theta)^2 t^2 \kappa\right) \sigma \, (h + l \sin(\theta))}{\kappa E t^3 \cos(\theta)} \tag{5.66}$$

which yields the elastic modulus in the radial direction as

$$Er := \frac{\kappa E t^3 \cos(\theta)}{\left(\sin(\theta)^2 \kappa l^2 + 2\sin(\theta)^2 t^2 + 2\sin(\theta)^2 t^2 \nu + \cos(\theta)^2 t^2 \kappa\right)(h + l\sin(\theta))}$$

$$= \frac{E t^3 \cos(\theta)}{(h + l\sin(\theta))\, l^2 \sin(\theta)^2 \left(1 + \frac{\left(2\frac{1}{\kappa} + 2\frac{\nu}{\kappa} + \cot(\theta)^2\right)t^2}{l^2}\right)} \tag{5.67}$$

or if κ is substituted as 5/6 yields,

$$E_r = \frac{E t^3 \cos(\theta)}{(h + l\sin(\theta))\, l^2 \sin(\theta)^2 \left(1 + \frac{\left(\frac{12}{5} + \frac{12}{5}\nu + \cot(\theta)^2\right)t^2}{l^2}\right)} \tag{5.68}$$

Equation (5.68) compares well with Gibson's advanced equation (5.13), only the term in the denominator, 2.4ν is replaced by 1.5ν. This is because the theoretical expression for displacement in equation (5.64) over estimates the deflection as observed by experimental result. Gibson's advanced equation was derived from deflection taken from Roarks [52] which was adjusted for the theoretical deficiency.

Strain in the tangential direction due to Poisson effect is

$$\varepsilon_{rt} = \delta 11 / (h + l\sin(\theta))$$

$$= \frac{\left(\kappa l^2 + 2t^2 + 2t^2\nu - t^2\kappa\right)\sin(\theta)\cos(\theta)\, l\, \sigma}{\kappa E t^3} \tag{5.69}$$

$$\nu_{rt} = \varepsilon_{rt}/\varepsilon_r$$

$$= \frac{\left(\kappa l^2 + 2t^2 + 2t^2\nu - t^2\kappa\right)\sin(\theta)\cos(\theta)^2\, l}{\left(\sin(\theta)^2 \kappa l^2 + 2\sin(\theta)^2 t^2 + 2\sin(\theta)^2 t^2 \nu + \cos(\theta)^2 t^2 \kappa\right)(h + l\sin(\theta))}$$

$$= \frac{l\left(1+2\frac{t^2}{l^2\kappa}+2\frac{t^2\nu}{\kappa l^2}-\frac{t^2}{l^2}\right)\cos(\theta)^2}{\sin(\theta)\,(h+l\sin(\theta))\left(1+2\frac{t^2}{l^2\kappa}+2\frac{t^2\nu}{\kappa l^2}+\frac{\cot(\theta)^2 t^2}{l^2}\right)} \tag{5.70}$$

$$= \frac{l\left(1.+\frac{(1.4+2.4\,\nu)\,t^2}{l^2}\right)\cos(\theta)^2}{\sin(\theta)\,(h+l\sin(\theta))\left(1.+\frac{\left(2.4+2.4\,\nu+\cot(\theta)^2\right)t^2}{l^2}\right)} \tag{5.70A}$$

In the tangential load case, the displacement δ11 is found to be

$$\delta 11 := -\sigma l\cos(\theta)\left(2l\cos(\theta)^2t^2\nu+4t^2\cos(\theta)^2\sin(\theta)^2h\kappa+t^2l\sin(\theta)^2\kappa+2l\cos(\theta)^2t^2 +2t^2\cos(\theta)^4\kappa h+2t^2\sin(\theta)^4h\kappa+\kappa l^3\cos(\theta)^2\right)/\left(Et^3\kappa\right) \tag{5.71}$$

giving the elastic modulus in the tangential direction as

$$Et := \frac{Et^3\,(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\frac{\left(2\frac{1}{\kappa}+2\frac{\nu}{\kappa}+\tan(\theta)^2+2\frac{h}{l\cos(\theta)^2}\right)t^2}{l^2}\right)} \tag{5.72}$$

or if κ is substituted as 5/6 yields,

$$E_t = \frac{Et^3\,(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\frac{\left(\frac{12}{5}+\frac{12}{5}\nu+\tan(\theta)^2+2\frac{h}{l\cos(\theta)^2}\right)t^2}{l^2}\right)} \tag{5.73}$$

Again equation (5.73) compares well with Gibson's advanced equation (5.14), only the term in the denominator, 2.4ν is replaced by 1.5ν. This is because the theoretical deficiency as explained earlier.

For shear modulus, the solution yields δ10 as;

$$\delta 10 := \left(2\,\kappa\, l\cos(\theta)^2 h^3 + \kappa\cos(\theta)^2 l^2 h^2 + \sin(\theta)^2 \kappa\, h^2 t^2 + 2\sin(\theta)\, t^2 \kappa\, l\, h + t^2 \kappa\, l^2 + 2\cos(\theta)^2 t^2 h^2 + 2\cos(\theta)^2 t^2 \nu\, h^2 + 4\, h\, l\cos(\theta)^2 t^2 + 4\, h\, l\cos(\theta)^2 t^2 \nu\right)\sigma \Big/ \left(\kappa\cos(\theta)\, t^3 E\right) \tag{5.74}$$

and the shear modulus

$$G_{rt} = \frac{Et^3(h + l\sin(\theta))}{h^2 l\cos(\theta)(l+2h)\left(1 + \dfrac{t^2(l + h\sin(\theta))^2}{h^2 l(l+2h)\cos(\theta)^2} + \dfrac{2t^2(2l+h)(1+\nu)}{\kappa hl(l+2h)}\right)} \tag{5.73}$$

or if κ is substituted as 5/6 yields,

$$G_{rt} = \frac{Et^3(h + l\sin(\theta))}{h^2 l\cos(\theta)(l+2h)\left(1 + \dfrac{t^2(l + h\sin(\theta))^2}{h^2 l(l+2h)\cos(\theta)^2} + \dfrac{2.4t^2(2l+h)(1+\nu)}{hl(l+2h)}\right)} \tag{5.76}$$

which gives the Gibson's advanced equation for shear modulus.

The derivations of section 5.3 and section 5.4 show that the Finite Element method give results that are comparable to the Gibson's result and validates the FE method as a suitable method for modelling the hexagonal wood cell models. The next section will expand the analytical FE model to use plate elements instead of the beam elements. In reality, the wood cells are plate structures rather than beam structures.

Plate elements will enable the third dimension of the wood cells to be modelled which are capable of sustaining loads both in bending and in-plane (membrane). The plate and shell type elements are much more difficult to satisfy in the compatibility requirements. The plate type element models will neglect the compatibility requirements at inter-element boundaries, these are more demanding than in the continuum case as complex coupled degree of freedoms interact with one another.

5.5 Analytical solution using plate with bending and membrane force

The FEM analysis of Wood cell using beam element in earlier section is extended to isotropic plate element. Wood cells are more accurately represented as a plate structure than beam structure. Plate element, which carry lateral loads in bending, are two dimensional analogues of the beam element.

2 possible cases of isotropic plate element are considered, with and without shear deformation. Stress in the longitudinal direction of the wood cell is considered as zero and no interference in radial or tangential direction as far as stress is concerned.

The basic model for the simple beam element is used, the element stiffness matrix is changed to the appropriate element stiffness matrix for plate element. Note that to distinguish the wall elastic properties, E_w is used instead of E in the plate element model.

5.5.1 Derivation of Plate element stiffness Matrix without Shear deformation

The Kirchhoff displacement approach is the most straightforward of the thin plate formulations. Element formulated are based on a single interpolation of the lateral displacement of the neutral plane. Shear strain energy is ignored and the resulting element is equivalent to the simple beam in the two dimensional model. The assumption is that straight lines normal to the mid-surface remain straight and normal to that surface after deformation.

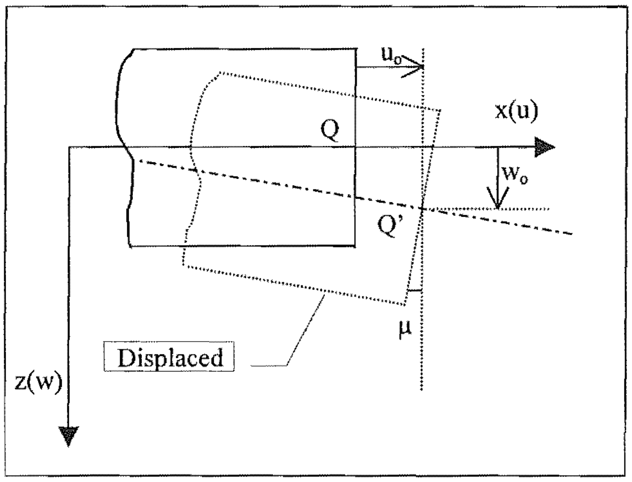


Figure 5.8 Plate element deformation

A geometrical construction in Figure 5.8 gives *u(x,y,z)*, *v(x,y,z)* and *w(x,y,z)*, the displacements in the cartesian coordinates within the plate as;

$$
\begin{aligned}
u(x,y,z) &= u_o(x,y) - z\frac{\partial w_o}{\partial x} \\
v(x,y,z) &= v_o(x,y) - z\frac{\partial w_o}{\partial y} \qquad (5.76) \\
w(x,y,z) &= w_o(x,y)
\end{aligned}
$$

The strain-displacement equations can be applied to gives the strain components;

$$
\begin{aligned}
e_x &= \frac{\partial u}{\partial x} = \frac{\partial u_o}{\partial x} - z\frac{\partial^2 w_o}{\partial x^2} \\
e_y &= \frac{\partial v}{\partial y} = \frac{\partial v_o}{\partial y} - z\frac{\partial^2 w_o}{\partial y^2} \qquad (5.77) \\
\gamma_{xy} &= \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} = \frac{\partial u_o}{\partial y} + \frac{\partial v_o}{\partial x} - 2z\frac{\partial^2 w_o}{\partial x \partial y}
\end{aligned}
$$

Each of these have two distinct contributions; the first involves in-plane derivatives of u_o and v_o arising from stretching and shearing of the centroidal plane. The second is the consequence of lateral (bending) deformation and varies linearly with depth, being proportional to the curvatures of the centroidal plane represented by the second derivatives. These 'membrane' and 'bending' contributions are denoted by 'o' and 'b' respectively. Using these notations, equation (5.77) can be written as;

$$
\begin{aligned}
e_x &= e_x^o + e_x^b \\
e_y &= e_y^o + e_y^b \\
\gamma_{xy} &= \gamma_{xy}^o + \gamma_{xy}^b
\end{aligned}
\tag{5.78}
$$

where $e_x^o = \frac{\partial u_o}{\partial x}, e_x^b = -z\frac{\partial^2 w_o}{\partial x^2}$ and $\gamma_{xy}^b = \text{-}2z\frac{\partial^2 w_0}{\partial x \partial y}$ etc, assuming that the through-thickness direct stress, σ_z is small in comparison to the in-plane stresses. From Hooke's law and with some manipulation yields; membrane stresses and bending stresses;

$$
\begin{aligned}
\sigma_x^o &= \frac{E}{1-\nu^2}\left(e_x^o + \nu e_y^o\right) \\
\sigma_y^o &= \frac{E}{1-\nu^2}\left(e_y^o + \nu e_x^o\right) \\
\tau_{xy}^o &= \frac{E}{2(1+\nu)}\gamma_{xy}^o
\end{aligned}
\tag{5.79}
$$

$$
\begin{aligned}
\sigma_x^b &= \frac{E}{1-\nu^2}\left(e_x^b + \nu e_y^b\right) \\
\sigma_y^b &= \frac{E}{1-\nu^2}\left(e_y^b + \nu e_x^b\right) \\
\tau_{xy}^b &= \frac{E}{2(1+\nu)}\gamma_{xy}^b
\end{aligned}
\tag{5.80}
$$

The membrane stresses (5.79) depends on the membrane strains and are constant through out the thickness of the plate. The bending stresses (5.80) vary linearly with the depth and gives a zero force resultant but have moments about the centroidal plane. The equaivalent moments per unit length are given by the integral across the depth;

$$
\begin{aligned}
M_x &= \int_{-t/2}^{t/2} \sigma_x z dz \\
M_y &= \int_{-t/2}^{t/2} \sigma_y z dz \\
M_{xy} &= \int_{-t/2}^{t/2} \tau_{yx} z dz
\end{aligned}
\tag{5.81}
$$

Substituting equation (5.79) and (5.80) into (5.81) yields;

$$M_x = -D\left(\frac{\partial^2 w_o}{\partial x^2} + \nu\frac{\partial^2 w_0}{\partial y^2}\right)$$
$$M_y = -D\left(\frac{\partial^2 w_o}{\partial y^2} + \nu\frac{\partial^2 w_0}{\partial x^2}\right) \qquad (5.82)$$
$$M_{xy} = -D(1-\nu)\frac{\partial^2 w_o}{\partial x \partial y}$$

where D is the flexural stiffness or rigidity of the plate given by $D = \dfrac{Et^3}{12(1-\nu^2)}$. These equations relate the 'moment-curvature' relationships for an isotropic elastic plate. If the distributed moments are known, the curvatures can be obtained. Unfortunately, the moments cannot be obtained using straightforward equilibrium method but only as solution of coupled partial differential equations involving the distributed moments, the distributed shear forces and distributed load on the surface of the plate. The energy method will be applicable for wood cell plate model in order to circumvent the complexity of solving the differential equations.

The equation (5.82) can be re-written in matrix form;

$$\begin{bmatrix} M_x \\ M_y \\ M_{xy} \end{bmatrix} = \begin{bmatrix} D & \nu D & 0 \\ \nu D & D & 0 \\ 0 & 0 & \dfrac{D}{2(1-\nu)} \end{bmatrix} \begin{bmatrix} -\dfrac{\partial^2 w}{\partial x^2} \\ -\dfrac{\partial^2 w}{\partial y^2} \\ -\dfrac{2\partial^2 w}{\partial x \partial y} \end{bmatrix} \qquad (5.83)$$

where $D = \dfrac{Et^3}{12(1-\nu^2)}$. This equation can be written as

$$[s] = [D][e] \qquad (5.84)$$

The strain energy per unit area of a plate is obtained by integrating the strain energy per unit volume through the thickness of the plate defined by the stresses σ_x, σ_y and τ_{xy}.

$$SE/area = \int_{-t/2}^{t/2} \frac{1}{2}\left(\sigma_x e_x + \sigma_y e_y + \tau_{xy}\gamma_{xy}\right)dz \qquad (5.85)$$

Equation (5.78) can be written as;

$$\sigma_x = \sigma_x^o + z\frac{12M_x}{t^3}$$

$$\sigma_y = \sigma_{yx}^o + z\frac{12M_y}{t^3} \qquad (5.86)$$

$$\tau_{xy} = \tau_{xy}^o + z\frac{12M_{xy}}{t^3}$$

Substituting equation (5.86) and (5.78) gives;

$$SE / area = \frac{1}{2}t\left(\sigma_x^o e_x^o + \sigma_y^o e_y^o + \tau_{xy}^o \gamma_{xy}^o\right)$$

$$+\frac{1}{2}\left(M_x\left[-\frac{\partial^2 w_o}{\partial x^2}\right]+M_y\left[-\frac{\partial^2 w_o}{\partial y^2}\right]+2M_{xy}\left[-\frac{\partial^2 w_o}{\partial x \partial y}\right]\right) \qquad (5.87)$$

The first term is the in-plane membrane stress strain energy and the second term the strain energy of the bending effects.

A major problem of Kirchhoff plate element is the difficulty encountered in satisfying the compatibility requirements at inter-element boundaries. The continuity of a plate element requires that the displacement w is continuous but the derivatives of w with respect to x and y are continuous as well, kinks at inter-element boundaries are normally encountered. Fortunately for the wood cell models, the inter-element boundaries are normally not continuous but are jointed at an angle and the impact of the compatibility problem may not be significant to affect the results.

A general expression for the stiffness matrix for a Kirchhoff plate element is now derived with the shape function in equation (4.27). The element stiffness matrix per unit depth $[K^e]$ from equation (4.19) is now integrated over the cross-sectional area of the element instead of the volume.

$$[K^e] = \int_A ([B^e]^T [D] \; [B^e]) \, dA \qquad (5.88)$$

where the strain-displacement matrix is given by $[B^e]$

$$[B^e]=\begin{bmatrix} -\frac{\partial^2 n_1}{\partial x^2} & -\frac{\partial^2 n_2}{\partial x^2} & -\frac{\partial^2 n_3}{\partial x^2} & -\frac{\partial^2 n_4}{\partial x^2} \\ -\frac{\partial^2 n_1}{\partial y^2} & -\frac{\partial^2 n_2}{\partial y^2} & -\frac{\partial^2 n_3}{\partial y^2} & -\frac{\partial^2 n_4}{\partial y^2} \\ -\frac{2\partial^2 n_1}{\partial x \partial y} & -\frac{2\partial^2 n_2}{\partial x \partial y} & -\frac{2\partial^2 n_3}{\partial x \partial y} & -\frac{2\partial^2 n_4}{\partial x \partial y} \end{bmatrix} \quad (5.89)$$

Evaluating matrix $[B^e]$ using the shape function of equation (4.27) yields,

$$\left[B^e\right]=\begin{bmatrix} -12\frac{x}{l^3}+6\frac{1}{l^2} & -6\frac{x}{l^2}+4\frac{1}{l} & -6\frac{1}{l^2}+12\frac{x}{l^3} & -6\frac{x}{l^2}+2\frac{1}{l} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (5.90)$$

Integrating the matrix, the element stiffness matrix $[K^e]$ gives the components;

$$\begin{bmatrix} -\frac{Ewt}{L^3(-1+v^2)} & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & \frac{Ewt}{L^3(-1+v^2)} & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} \\ -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{3}\frac{Ewt}{L(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{6}\frac{Ewt}{L(-1+v^2)} \\ \frac{Ewt}{L^3(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{Ewt}{L^3(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} \\ -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{6}\frac{Ewt}{L(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{3}\frac{Ewt}{L(-1+v^2)} \end{bmatrix} \quad (5.91)$$

Expanding the matrix to include membrane stress gives the general stiffness matrix

$$\begin{bmatrix} K & 0 & 0 & -K & 0 & 0 \\ 0 & -\frac{Ewt}{L^3(-1+v^2)} & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & 0 & \frac{Ewt}{L^3(-1+v^2)} & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} \\ 0 & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{3}\frac{Ewt}{L(-1+v^2)} & 0 & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{6}\frac{Ewt}{L(-1+v^2)} \\ -K & 0 & 0 & K & 0 & 0 \\ 0 & \frac{Ewt}{L^3(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & 0 & -\frac{Ewt}{L^3(-1+v^2)} & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} \\ 0 & -\frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{6}\frac{Ewt}{L(-1+v^2)} & 0 & \frac{1}{2}\frac{Ewt}{L^2(-1+v^2)} & -\frac{1}{3}\frac{Ewt}{L(-1+v^2)} \end{bmatrix} \quad (5.92)$$

which can be recast into a similar form to equation (5.20) where $K = \frac{Ewt}{L(1-v^2)}$, a=D/L^3 and $D = \frac{Ewt^3}{12(1-v^2)}$, D being the plate stiffness per unit length.

$$\begin{bmatrix} K & 0 & 0 & -K & 0 & 0 \\ 0 & 12\,a & 6\,a\,L & 0 & -12\,a & 6\,a\,L \\ 0 & 6\,a\,L & 4\,a\,L^2 & 0 & -6\,a\,L & 2\,a\,L^2 \\ -K & 0 & 0 & K & 0 & 0 \\ 0 & -12\,a & -6\,a\,L & 0 & 12\,a & -6\,a\,L \\ 0 & 6\,a\,L & 2\,a\,L^2 & 0 & -6\,a\,L & 4\,a\,L^2 \end{bmatrix} \tag{5.93}$$

Equation (5.93) can also be derived by letting $E = \frac{Ew}{(1-v^2)}$ in equation (5.20).

By imposing similar boundary condition as in section 5.3 and solving the energy equation (4.49) gives the various elastic modulus and shear modulus,

$$E_r = \frac{\left(\frac{Ew}{(1-v^2)}\right) t^3 \cos(\theta)}{l^2 \sin(\theta)^2 (h + l\sin(\theta)) \left(1 + \frac{t^2 \cot(\theta)^2}{l^2}\right)} \tag{5.94}$$

$$E_t = \frac{\left(\frac{Ew}{(1-v^2)}\right) t^3 (h + l\sin(\theta))}{l^4 \cos(\theta)^3 \left(1 + \frac{2t^2 h}{l^3 \cos(\theta)^2} + \frac{t^2 \tan(\theta)^2}{l^2}\right)} \tag{5.95}$$

$$G_{rt} = \frac{\left(\frac{Ew}{(1-v^2)}\right) t^3 (h + l\sin(\theta))}{h^2 l \cos(\theta)(2h + l)\left(1 + \frac{t^2 (l + h\sin(\theta))^2}{h^2 (2h + l) l \cos(\theta)^2}\right)} \tag{5.96}$$

$$v_{rt} = -\frac{l\cos(\theta)^2 \sin(\theta)\left(\frac{t^2}{l^2} - 1\right)}{(h + l\sin(\theta))\sin(\theta)\left(1 + \frac{t^2 \cot(\theta)^2}{l^2}\right)} \tag{5.97}$$

5.6 Analytical solution using plate with bending, membrane and shear effect.

The plate element stiffness matrix with the above consideration is similar to equation (4.47) expanded to include membrane effect,

$$[K^e]=\begin{bmatrix} K & 0 & 0 & -K & 0 & 0 \\ 0 & 12a & 6aL & 0 & -12a & 6aL \\ 0 & 6aL & (4+e)aL^2 & 0 & -6aL & (2-e)aL^2 \\ -K & 0 & 0 & K & 0 & 0 \\ 0 & -12a & -6aL & 0 & 12a & -6aL \\ 0 & 6aL & (2-e)aL^2 & 0 & -6aL & (4+e)aL^2 \end{bmatrix} \tag{5.98}$$

where E is replaced by $E=\dfrac{E_w}{(1-v^2)}$, a= D/L^3, D is the flexural stiffness of the plate and $e=\dfrac{12D}{\kappa G t l^2}$, κ=5/6.

Solving the energy equation (4.19) yields the elastic modulus;

$$E_r=\frac{\left(\dfrac{E_w}{1-v^2}\right)t^3\cos(\theta)}{(h+l\sin(\theta))l^2\sin(\theta)^2\left(1+\dfrac{t^2\cot(\theta)^2}{l^2}+\dfrac{2.4t^2}{l^2(1-v)}\right)} \tag{5.99}$$

$$E_t=\frac{\left(\dfrac{E_w}{1-v^2}\right)t^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(\left(\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)\dfrac{t^2}{l^2}+\dfrac{2.4t^2}{l^2(1-v)}\right)} \tag{5.100}$$

$$\nu_{rt} = \frac{l\left(1+\frac{(1.4+\nu)t^2}{(1-\nu)l^2}\right)\cos(\theta)^2}{\sin(\theta)(h+l\sin(\theta))\left(1+\frac{\left(2.4+2.4\nu+\cot(\theta)^2\right)t^2}{l^2}\right)} \tag{5.101}$$

$$G_{rt} = \frac{\left(\frac{E_w}{1-\nu^2}\right)t^3(h+l\sin(\theta))}{h^2 l\cos(\theta)(l+2h)\left(1+\frac{t^2\left(l+h\sin(\theta)\right)^2}{lh^2(l+2h)\cos(\theta)^2}+2.4\frac{t^2(h+2l)}{lh(l+2h)(1-\nu)}\right)} \tag{5.102}$$

5.7 Analytical solution using plate element with bending, membrane and shear effect and longitudinal strain effect.

A refinement to the plate element model is to consider the Poisson's effect in the longitudinal direction. When the wood cell is stressed in the R or T direction, Poisson's effect will induce a strain in the L direction proportional to the in-plane strain. Though the walls of the cell are subjected to different strains in the tangential and radial directions, compatibility requires there be one strain e_o in the longitudinal direction at least at the boundaries. If we assumed elastic deformation and local plastic deformations are not allowed, the deformation e_o will be uniform in the longitudinal direction.

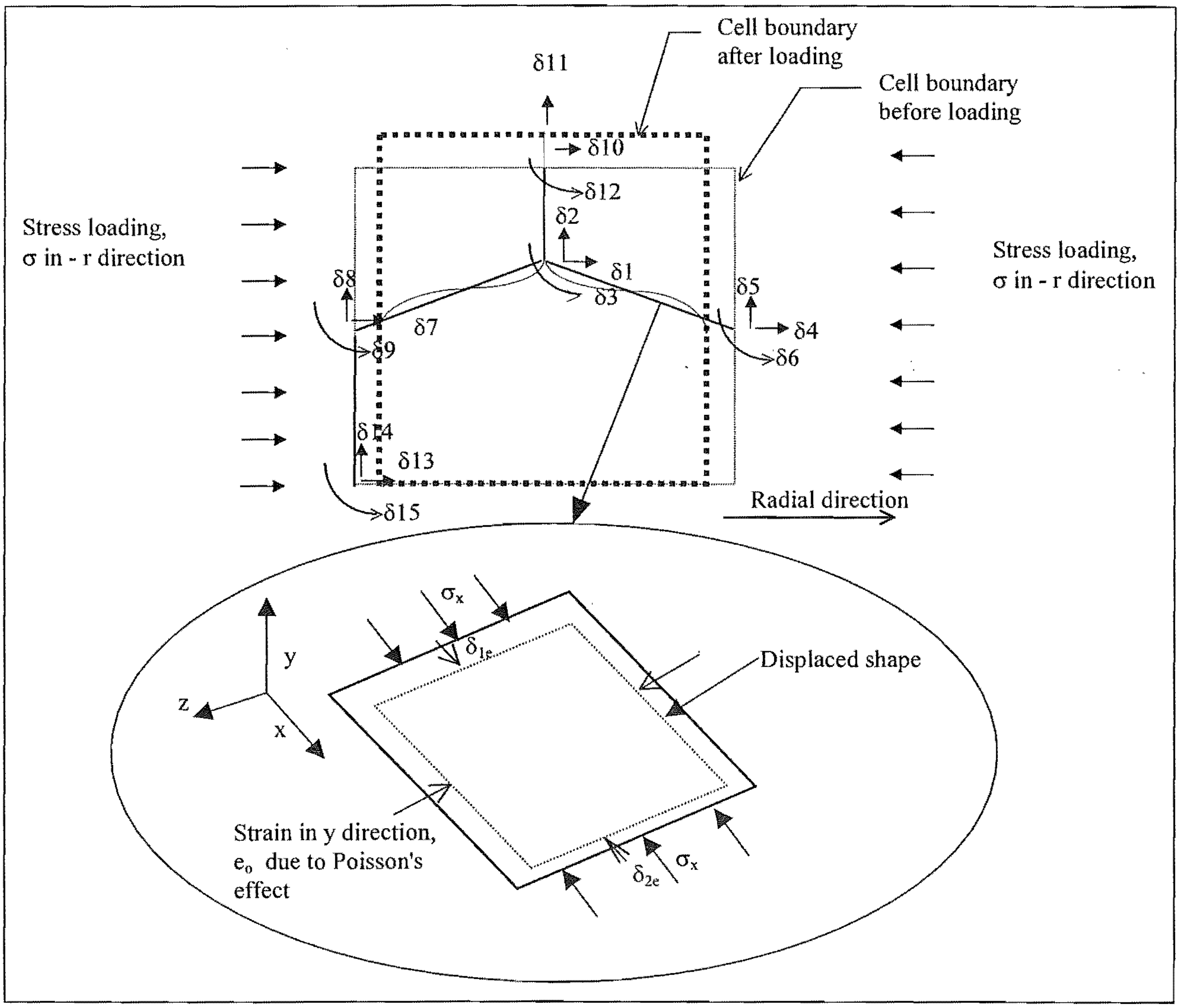


Figure 5.8 Poisson's effect on wood cell longitudinal strain

Consider a three dimensional wood cell loaded with stress σ_x in the x direction (radial direction) and because of Poisson's effect, there will be a strain in the longitudinal direction.

The strain energy of the plate would be given by:

$$\text{Strain energy / unit volume} = \tfrac{1}{2}\,[\sigma_x e_x + \sigma_y e_o] \tag{5.103}$$

where

$$\sigma_x = \frac{E_w(e_x + v_{eo})}{1-v^2} \tag{5.104}$$

$$\sigma_y = \frac{E_w(e_o + v_{ex})}{1-v^2} \tag{5.105}$$

$$e_x = \frac{\delta 1_e - \delta 2_e}{l} \tag{5.106}$$

and e_o is the strain in the y (longitudinal) direction and the subscript 'e' means the element in-plane displacements in the local co-ordinates. This must be transform to the global co-ordinates when evaluating the equations in the global form.

Expanding the energy equation,

$$\text{SE / unit volume} = \frac{1}{2}\frac{E_w(e_x^{\,2} + 2ve_xe_o + e_o^{\,2})}{1-v^2}$$

(5.107)

$$\text{SE for a cell wall} = \frac{1}{2}\frac{E_w(\delta 2_e - \delta 1_e)^2 A}{(1-v^2)l} - \frac{v_{eo}E_w(\delta 2_e - \delta 1_e)A}{(1-v^2)} + \frac{1}{2}\frac{E_w A l e_o^{\,2}}{(1-v^2)} \tag{5.108}$$

Re-writing in stiffness matrix form,

$$\text{SE} = \frac{1}{2}\left[d^T\right]\begin{bmatrix} K' & -K' \\ -K' & K' \end{bmatrix}[d] + \left[d^T\right]\begin{bmatrix} \dfrac{E_w e_o A}{1-v^2} \\ \dfrac{-E_w e_o A}{1-v^2} \end{bmatrix} v + \frac{1}{2}K' e_o^{\,2} l^2 \qquad (5.109)$$

where $K' = \dfrac{E_w A}{(1-v^2)l}$ and $d = \begin{bmatrix} \delta 1_e \\ \delta 2_e \end{bmatrix}$

The total energy of the cell wall plates is then given by,

$$\chi = \frac{1}{2}\left[\delta^T\right][K][\delta] + e_o v\left[d^T\right]\left[F'_{eo}\right] + \Sigma\frac{1}{2}K' e_o^{\,2} l^2 - \left[\delta^T\right][F] \qquad (5.110)$$

where $[F'_{eo}] = \begin{bmatrix} E_w A/(1-v^2) \\ -E_w A/(1-v^2) \end{bmatrix}$

The first term is due to the strain energy of the plate, including the membrane, shear and bending effect. The second and third terms are due to the e_o strain in the longitudinal direction and the last term due to the potential energy of the applied stress.

Differentiating the total energy equation (5.110),

$$\frac{\partial \chi}{\partial \delta_i} = [K]\,[d] = [F] + e_o v\,[F'_{eo}] \qquad (5.111)$$

$$\frac{\partial \chi}{\partial e_o} = v\,[F'_{eo}]\,[\,d\,] \qquad (5.112)$$

$$= v\,[\,d^T\,]\,[F'_{eo}] = -\Sigma K' l^2 e_o = \alpha e_o \qquad (5.113)$$

where $\alpha = \Sigma K' l^2$

From equation (5.113),

$$e_o = \frac{v}{\alpha}\left[d^T\right]\left[F'_{eo}\right] \qquad (5.114)$$

Substituting e_o into equation (5.111),

$$[K]\,[d] = [F] + \frac{v}{\alpha}[F'_{eo}]\,[F'^{T}_{eo}]\,[d] \qquad (5.115)$$

$$[K - \frac{v}{\alpha} K'_{eo}]\,[d] = [F] \qquad (5.116)$$

where $K'_{eo} = [F'_{eo}]\,[F'^{T}_{eo}]$

5.7.1 Elastic modulus

The boundary conditions and coupled nodes are similar to the earlier models in section 5. To evaluate the function α and $[F'_{eo}]$,

$$\begin{aligned}\alpha &= \Sigma K' l^2 \\ &= \sum \frac{EwAl}{1-v^2} \\ &= \frac{Ewt(2l+h)}{1-v^2}\end{aligned} \qquad (5.117)$$

$$[F'_{eo}] = \begin{bmatrix} \dfrac{Ewt}{1-v^2} \\ -\dfrac{Ewt}{1-v^2} \end{bmatrix} \qquad (5.118)$$

where $A = t*1$ if we consider a cell of depth of unity.

Therefore $K'_{eo} = [F'_{eo}]\,[F'^{T}_{eo}]$

$$= \begin{bmatrix} -\dfrac{Ew\,t}{(2l+h)(-1+v^2)} & \dfrac{Ew\,t}{(2l+h)(-1+v^2)} \\ \dfrac{Ew\,t}{(2l+h)(-1+v^2)} & -\dfrac{Ew\,t}{(2l+h)(-1+v^2)} \end{bmatrix} \qquad (5.119)$$

Combining the effect of e_o with the element stiffness matrix from equation (5.98) gives the element stiffness matrix of the element,

$$\begin{bmatrix} K+\frac{\nu\, Ew\, t}{(2l+h)\left(\nu^2-1\right)} & 0 & 0 & -K-\frac{\nu\, Ew\, t}{(2l+h)\left(\nu^2-1\right)} & 0 & 0 \\ 0 & 12\,a & 6\,a\,L & 0 & -12\,a & 6\,a\,L \\ 0 & 6\,a\,L & (4+e)\,a\,L^2 & 0 & -6\,a\,L & (2-e)\,a\,L^2 \\ -K-\frac{\nu\, Ew\, t}{(2l+h)\left(\nu^2-1\right)} & 0 & 0 & K+\frac{\nu\, Ew\, t}{(2l+h)\left(\nu^2-1\right)} & 0 & 0 \\ 0 & -12\,a & -6\,a\,L & 0 & 12\,a & -6\,a\,L \\ 0 & 6\,a\,L & (2-e)\,a\,L^2 & 0 & -6\,a\,L & (4+e)\,a\,L^2 \end{bmatrix} \tag{5.120}$$

where $K = \dfrac{Et}{L(1-\nu^2)}$, $a = D/L^3$, D being the flexural stiffness of the plate and

$e = \dfrac{12D}{\kappa G t l^2}$, $\kappa = 5/6$.

The energy equation (5.110) can be evaluated and solved using the energy method to yield the elastic modulus and shear modulus,

$$E_r = \frac{\left(\frac{E_w}{1-\nu^2}\right)t^3\cos(\theta)}{l^2\sin(\theta)^2\left(h+l\sin(\theta)\right)\left(1+\frac{t^2\cot(\theta)^2(2l+h)}{l^2(2l+h-\nu l)}+\frac{2.4t^2}{(1-\nu)l^2}\right)} \tag{5.121}$$

$$E_t = \frac{\left(\frac{E_w}{1-\nu^2}\right)t^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\frac{t^2(2l+h)}{l^2(2l+h-\nu l)}\left(\tan(\theta)^2+\frac{2h}{l\cos(\theta)^2}\right)+\frac{2.4t^2}{(1-\nu)l^2}\right)} \tag{5.122}$$

$$\nu_{rt} = \frac{l\cos(\theta)^2\left(1+\frac{t^2}{l^2}\left(\frac{2.4}{1-\nu}-\frac{2l+h}{h+2l-\nu l}\right)\right)}{\sin(\theta)(h+l\sin(\theta))\left(1+\frac{t^2}{l^2}\left(\frac{2.4}{1-\nu}+\frac{\cot(\theta)^2(2l+h)}{h+2l-\nu l}\right)\right)} \tag{5.123}$$

$$G_{rt} = \frac{\left(\frac{E_w}{1-\nu^2}\right)t^3(h+l\sin(\theta))}{h^2 l\cos(\theta)(l+2h)\left(1+\frac{t^2\left(h+l\sin(\theta)\right)^2(2l+h)}{h^2(l+2h)(2l+h-\nu l)l\cos(\theta)^2}+\frac{2.4t^2(2l+h)}{hl(1-\nu)(l+2h)}\right)} \tag{5.124}$$

5.8 Comparison of analytical solutions

The analytical solutions of the regular hexagonal wood cell are derived and developed from beam elements and then progressed to plate element using increasing complex considerations. As the models are developed in complexity, the effect of each considerations are readily visble.

These works resulted in 2 significant plate-type solutions;

1. Plate solution with membrane, bending and shear without coupled boundaries,
2. Plate solution with membrane, bending, shear and coupled boundaries, (i.e. Poisson's effect on longitudinal strain e_0 caused by coupling of plates boundaries to form a continuum).

The analytical solutions are found to agree at v=0 since longitudinal strain e_0 equal to zero in all cases and the influence of ν is being nullified by setting to zero. This is a quick check against agreement of the derived analytical solution to Gibson's equation.

The comparison of the beam solutions is tabulated in table 5.1 with a simple plate solution as a comparison. Gibson's simple equation takes into consideration only the bending stress, introduction of the membrane and shear factor resulted in the Gibson's simple equation over-estimating the elastic modulus. The effects of moment-curvature coupling is evident even in simple Kirchhoff plate element, the wall material elastic property is now modified by the term, $\frac{1}{1-\nu^2}$ as shown in column three of Table 5.1.

Plate model results are shown in Table 5.2. The plate with shear consideration has an addition term, $\frac{2.4t^2}{l^2(1-\nu)}$ caused by the shear effect compared to the Kirchhoff plate model. With the additional consideration of the e_0 strain, the membrane stress term is modified by the term, $\frac{2l+h}{2l+h-\nu l}$.

	Beam element with bending deformation [Gibson's simple equation]	Beam Element with bending and membrane stress	Beam element with bending, membrane stress and shear effect [Gibson's advanced equation]	Plate element with membrane and bending Stress.
Er	$\frac{E t^3 \cos(\theta)}{l^2 \sin(\theta)^2 (h+l\sin(\theta))}$	$\frac{E t^3 \cos(\theta)}{l^2 \sin(\theta)^2 \left(1+\frac{t^2\cot(\theta)^2}{l^2}\right)(h+l\sin(\theta))}$	$\frac{Et^3\cos(\theta)}{(h+l\sin(\theta))l^2\sin(\theta)^2\left(1+\frac{t^2}{l^2}\left(2.4+2.4\nu\cot(\theta)^2\right)\right)}$	$\frac{\left(\frac{Ew}{(1-\nu^2)}\right)t^3\cos(\theta)}{l^2\sin(\theta)^2(h+l\sin(\theta))\left(1+\frac{t^2\cot(\theta)^2}{l^2}\right)}$
Et	$\frac{t^3 E(h+l\sin(\theta))}{l^4\cos(\theta)^3}$	$\frac{t^3E(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+2\frac{t^2h}{l^3\cos(\theta)^2}+\frac{t^2\tan(\theta)^2}{l^2}\right)}$	$\frac{Et^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\frac{t^2}{l^2}\left(2.4+2.4\nu+\tan(\theta)^2+\frac{2h}{l\cos(\theta)^2}\right)\right)}$	$\frac{\left(\frac{Ew}{(1-\nu^2)}\right)t^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\frac{2t^2h}{l^3\cos(\theta)^2}+\frac{t^2\tan(\theta)^2}{l^2}\right)}$
Grt	$\frac{Et^3(h+l\sin\theta)}{lh^2\cos\theta(l+2h)}$	$\frac{t^3E(h+l\sin(\theta))}{h^2l\cos(\theta)(2h+l)\left(1+\frac{t^2(l+h\sin(\theta))^2}{h^2(l+2h)l\cos(\theta)^2}\right)}$	$\frac{Et^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\frac{t^2(l+h\sin(\theta))^2}{h^2l(l+2h)\cos(\theta)^2}+\frac{2.4t^2(2l+h)(1+\nu)}{hl(l+2h)}\right)}$	$\frac{\left(\frac{Ew}{(1-\nu^2)}\right)t^3(h+l\sin(\theta))}{h^2l\cos(\theta)(2h+l)\left(1+\frac{t^2(l+h\sin(\theta))^2}{h^2(2h+l)l\cos(\theta)^2}\right)}$
vrt	$-\frac{l\cos(\theta)^2}{(h+\sin(\theta)l)\sin(\theta)}$	$-\frac{l\cos(\theta)^2\left(\frac{t^2}{l^2}-1\right)}{(h+l\sin(\theta))\sin(\theta)\left(1+\frac{t^2\cot(\theta)^2}{l^2}\right)}$	$\frac{l\left(1.+\frac{(1.4+2.4\nu)t^2}{l^2}\right)\cos(\theta)^2}{\sin(\theta)(h+l\sin(\theta))\left(1.+\frac{(2.4+2.4\nu+\cot(\theta)^2)t^2}{l^2}\right)}$	$-\frac{l\cos(\theta)^2\sin(\theta)\left(\frac{t^2}{l^2}-1\right)}{(h+l\sin(\theta))\sin(\theta)\left(1+\frac{t^2\cot(\theta)^2}{l^2}\right)}$

Table 5.1. Beam elements

	Plate element with bending, membrane stress and shear deformation without eo strain	Plate element with bending, membrane and shear and eo strain
Er	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3\cos(\theta)}{(h+l\sin(\theta))l^2\sin(\theta)^2\left(1+\dfrac{t^2\cot(\theta)^2}{l^2}+\dfrac{2.4t^2}{l^2(1-v)}\right)}$	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3\cos(\theta)}{(h+l\sin(\theta))l^2\sin(\theta)^2\left(1+\dfrac{t^2\cot(\theta)^2(2l+h)}{l^2(2l+h-vl)}+\dfrac{2.4t^2}{(1-v)l^2}\right)}$
Et	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(\left(\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)\dfrac{t^2}{l^2}+\dfrac{2.4t^2}{l^2(1-v)}\right)}$	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\dfrac{t^2(2l+h)}{l^2(2l+h-vl)}\left(\tan(\theta)^2+\dfrac{2h}{l\cos(\theta)^2}\right)+\dfrac{2.4t^2}{(1-v)l^2}\right)}$
Grt	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\dfrac{t^2(l+h\sin(\theta))^2}{lh^2(l+2h)\cos(\theta)^2}+2.4\dfrac{t^2(h+2l)}{lh(l+2h)(1-v)}\right)}$	$\dfrac{\left(\dfrac{E_w}{1-v^2}\right)t^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\dfrac{t^2(h+l\sin(\theta))^2(2l+h)}{h^2(l+2h)(2l+h-vl)l\cos(\theta)^2}+\dfrac{2.4t^2(2l+h)}{hl(1-v)(l+2h)}\right)}$
νrt	$\dfrac{l\left(1+\dfrac{(1.4+v)t^2}{(1-v)l^2}\right)\cos(\theta)^2}{\sin(\theta)(h+l\sin(\theta))\left(1+\dfrac{(2.4+2.4v+\cot(\theta)^2)t^2}{l^2}\right)}$	$\dfrac{l\cos(\theta)^2\left(1+\dfrac{t^2}{l^2}\left(\dfrac{2.4}{1-v}-\dfrac{2l+h}{h+2l-vl}\right)\right)}{\sin(\theta)(h+l\sin(\theta))\left(1+\dfrac{t^2}{l^2}\left(\dfrac{2.4}{1-v}+\dfrac{\cot(\theta)^2(2l+h)}{h+2l-vl}\right)\right)}$

Table 5.2. Comparison of plate elements

Chapter 6.

Comparison of analytical solutions with Gibson's simple, advanced equation, plate elements and FEM models

The results of the plate and beam analytical solutions are compared with the FEM models with various Poisson's ratios and cellular angles. Regular FEM models using SHELL91 elements are constructed using cellular angle of zero, fifteen and thirty degrees and $\alpha=0$ (no offset). SHELL91 are multi-layered thick plate elements capable of supporting in-plane membrane and shear deformation. Cyclic constrains are applied to model boundary conditions using Stol's cyclic algorithm. (see appendix for mathematical treatment). The models are repeated with various Poisson's ratios and t/H ratios for each cellular angle and tabulated in the appendix, the results of the FEM models and the analytical are plotted for comparison.

Plate analytical solutions with e_o strain consideration are denoted with a subscript 'o' and plate solutions without e_o strain consideration are denoted by the subscript 'p'. The elastic moduli are normalized to Gibson's simple equation, denoted by subscript 's'; Gibson's advanced equation is denoted by subscript 'p'. The various moduli, Er, Et, Grt and Poisson's ratio vrt are plotted against t/H to trend the variation of the various elastic properties. The results show that the analytical plate solutions agree very closely with the FEM models. In some cases, the graphs of the FEM models and the plate solutions overlap each other.

The deficiency of Gibson's advanced model is clearly visible as the Poisson's ratio increase, the Gibson's model deviates significantly from the FEM model. The plate models when written in the forms as in Table 6.1 show that the plate solutions have two contributions, the first being the beam effect as considered by Gibson in her advanced equation, the second due to the plate coupling effect contributed by a factor $-\nu^2$ on the bending and membrane strain effects. For example the radial modulus for plate model without e_o strain is given by;

$$E_r = \frac{E w t^3 \cos(\theta)}{(h + l\sin(\theta))\, l^2 \sin(\theta)^2 \left(1 + \frac{\left(2.4 + 2.4\,\nu + \cot(\theta)^2\right) t^2}{l^2} - \nu^2 \left(1 + \frac{\cot(\theta)^2 t^2}{l^2}\right)\right)}$$

Beam effect,
Gibson advanced eqn.

Plate coupling effect on bending
and membrane strains.

	Beam element with bending, membrane stress and shear effect [Gibson's advanced equation]	Plate Element with membrane, bending and Shear effect, but without eo strain	Plate element with membrane, bending and Shear effect with eo strain.
Er	$\dfrac{Et^3\cos(\theta)}{(h+l\sin(\theta))l^2\sin(\theta)^2\left(1+\dfrac{t^2}{l^2}\left(2.4+2.4\nu\cot(\theta)^2\right)\right)}$	$\dfrac{Ewt^3\cos(\theta)}{(h+l\sin(\theta))\,l^2\sin(\theta)^2\left(1+\dfrac{\left(2.4+2.4\nu+\cot(\theta)^2\right)t^2}{l^2}-\nu^2\left(1+\dfrac{\cot(\theta)^2t^2}{l^2}\right)\right)}$	$\dfrac{Ewt^3\cos(\theta)}{(h+l\sin(\theta))\,l^2\sin(\theta)^2\left(1+\dfrac{\left(2.4+2.4\nu+\dfrac{\cot(\theta)^2(2l+h)}{2l+h-\nu l}\right)t^2}{l^2}-\nu^2\left(1+\dfrac{\cot(\theta)^2(2l+h)t^2}{(2l+h-\nu l)l^2}\right)\right)}$
Et	$\dfrac{Et^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\dfrac{t^2}{l^2}\left(2.4+2.4\nu+\tan(\theta)^2+\dfrac{2h}{l\cos(\theta)^2}\right)\right)}$	$\dfrac{Ewt^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\dfrac{\left(2.4+2.4\nu+\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)t^2}{l^2}-\nu^2\left(1+\dfrac{\left(\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)t^2}{l^2}\right)\right)}$	$\dfrac{Ewt^3(h+l\sin(\theta))}{l^4\cos(\theta)^3\left(1+\dfrac{\left(2.4+2.4\nu+\dfrac{\left(\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)(2l+h)}{2l+h-\nu l}\right)t^2}{l^2}-\nu^2\left(1+\dfrac{\left(\tan(\theta)^2+2\dfrac{h}{l\cos(\theta)^2}\right)(2l+h)t^2}{(2l+h-\nu l)l^2}\right)\right)}$
Grt	$\dfrac{Et^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\dfrac{t^2(l+h\sin(\theta))^2}{h^2l(l+2h)\cos(\theta)^2}+\dfrac{2.4t^2(2l+h)(1+\nu)}{hl(l+2h)}\right)}$	$\dfrac{Ewt^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\dfrac{t^2\left(\dfrac{(l+h\sin(\theta))^2}{l\cos(\theta)^2}+2.4\dfrac{h(h+2l)(1+\nu)}{l}\right)}{h^2(l+2h)}-\nu^2\left(1+\dfrac{t^2(l+h\sin(\theta))^2}{h^2(l+2h)l\cos(\theta)^2}\right)\right)}$	$\dfrac{Ewt^3(h+l\sin(\theta))}{h^2l\cos(\theta)(l+2h)\left(1+\dfrac{t^2\left(\dfrac{(h+2l)(l+h\sin(\theta))^2}{(2l+h-\nu l)l\cos(\theta)^2}+2.4\dfrac{h(h+2l)(1+\nu)}{l}\right)}{h^2(l+2h)}-\nu^2\left(1+\dfrac{t^2(h+2l)(l+h\sin(\theta))^2}{h^2(l+2h)(2l+h-\nu l)l\cos(\theta)^2}\right)\right)}$
νrt	$\dfrac{l\left(1.+\dfrac{(1.4+2.4\nu)t^2}{l^2}\right)\cos(\theta)^2}{\sin(\theta)\,(h+l\sin(\theta))\left(1.+\dfrac{\left(2.4+2.4\nu+\cot(\theta)^2\right)t^2}{l^2}\right)}$	$-\dfrac{l\left(1+\dfrac{(1.4+\nu)t^2}{(1-\nu)l^2}\right)\cos(\theta)^2}{\sin(\theta)\,(h+l\sin(\theta))\left(1+\dfrac{\left(2.4+2.4\nu+\cot(\theta)^2\right)t^2}{l^2}\right)}$	$\dfrac{l\left(1+\dfrac{\left(2.4\dfrac{1}{1-\nu}-\dfrac{2l+h}{h+2l-\nu l}\right)t^2}{l^2}\right)\cos(\theta)^2}{\sin(\theta)\,(h+l\sin(\theta))\left(1+\dfrac{\left(2.4\dfrac{1}{1-\nu}+\dfrac{\cot(\theta)^2(2l+h)}{h+2l-\nu l}\right)t^2}{l^2}\right)}$

Table 6.1 Comparison of Gibson advanced model and various plate models

At ν=0, the various models converged at a single solution; Gibson's advanced model, since the plate effect is no longer contributing in the plate model. The FEM model though shows deviation at low t/H, being due to the coarseness of the FEM mesh and the convergence of the solution. As ν increases, the plate coupling effect on the bending and membrane strain becomes more prominent and the plate model deviates from the Gibson's advanced model.

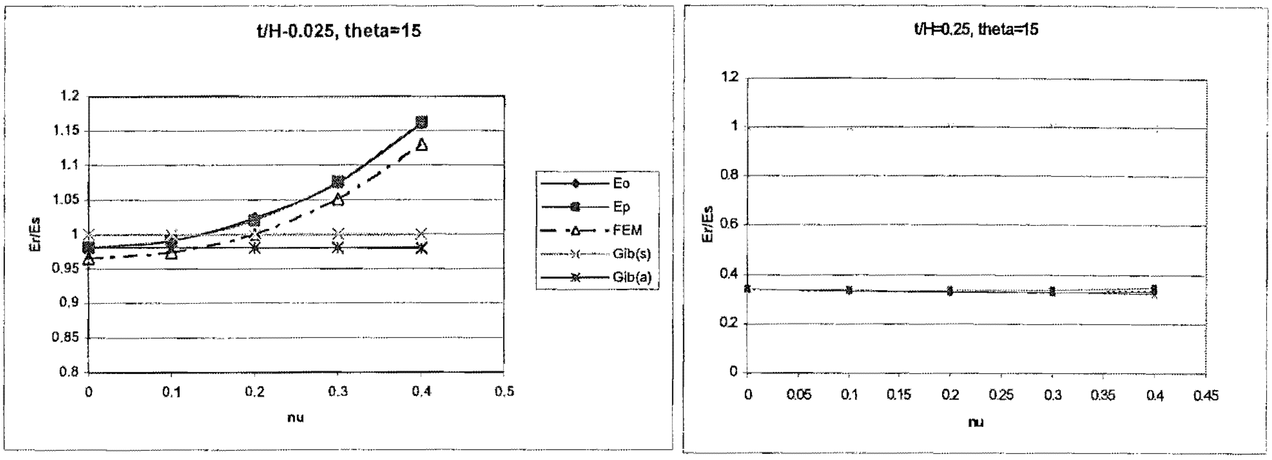


Figure 6.1 Effect of ν on t/H (0.025 and 0.25) for theta =15

However at high t/H values, the plate effect is less prominent because the thick wall cell is less 'plate' like and more rigid 'beam' like. The high plate stiffness makes the effect of plate coupling less significant and the plate model deviate less from Gibson's advanced model.

The graphs of various models with varying cell angles are show in Figure 6.2-6.50. Note that for cell angle equal to zero, the E_r results are singular and cannot be computed and therefore not shown. In all cell angle cases, plate-coupling effect dominates the solution at low t/H values (low flexural stiffness) and the Poisson's effect in the longitudinal direction is significant as the low flexural stiffness gives a larger e_0 strain, of course the magnitude of the e_0 strain is proportional to the Poisson's ratio. For example, for theta= 15, v=0 (figure 6.12), E_r for the analytical models coincide with each other as the plate model approaches Gibson advanced model. As v increases to 0.4, at low t/H (0.05), the E_0 and E_p models are deviates significantly from Gibson's advanced equation, but as t/H (0.25) increases the deviations decreases. Both plate models agree closely with the FEM model, the advance plate model with e_o strain provides a closer agreement to the FEM model.

This trend is evident for other mechanical properties; the beam and plate analytical solutions at ν=0 agrees very closely but as ν increases and at low t/H ratio, Gibson's advanced equation deviates as the plate-coupling effects becomes more significant. As ν increases, the plate effect becomes much more significant and Gibson's model fails while the plate solutions maintain its prediction very accurately; the e_o solution being in some cases co-incide with the FEM solution.

As t/H increases, the plate flexural strength increases and the model approaches a beam solution. This can be explained, when t/H increases, the flexural stiffness increases and transverse strain effect due to plate bending moment coupling and Poisson's effect are greatly reduced and the rigid plate approaches a beam type (Gibson's advanced) behavior. This is evident in the graphs as the deviation for the advanced beam solution and the plate solution are very close at high t/H value of 0.25.

The effect of e_o strain on the elastic modulus is quite insignificant compared to the simple plate solution, suggesting that the model stiffness in the longitudinal direction is very high compared to the radial and tangential directions and the coupling due to e_o effect does not affect the longitudinal strain significantly. The effect of e_o strain does not seems to affect the v_{rt} significantly either.

The results are in agreements with observed and measured elastic moduli in the E_r, E_t and E_l direction, the longitudinal direction being the stiffest with at least a ratio of 11:1. The wall coupling effect will have negligible effect on the longitudinal direction.

6.1 Graphs for analytical solutions, cell angle =0

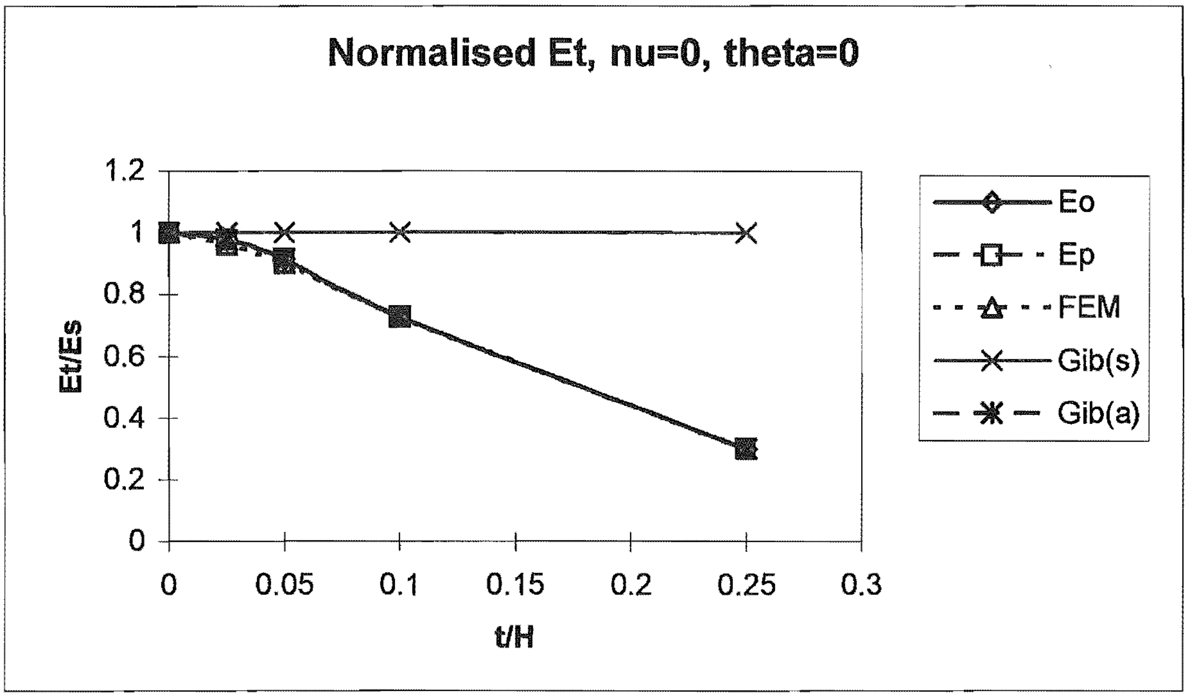


Figure 6.2 Et for cell angle =0, v=0

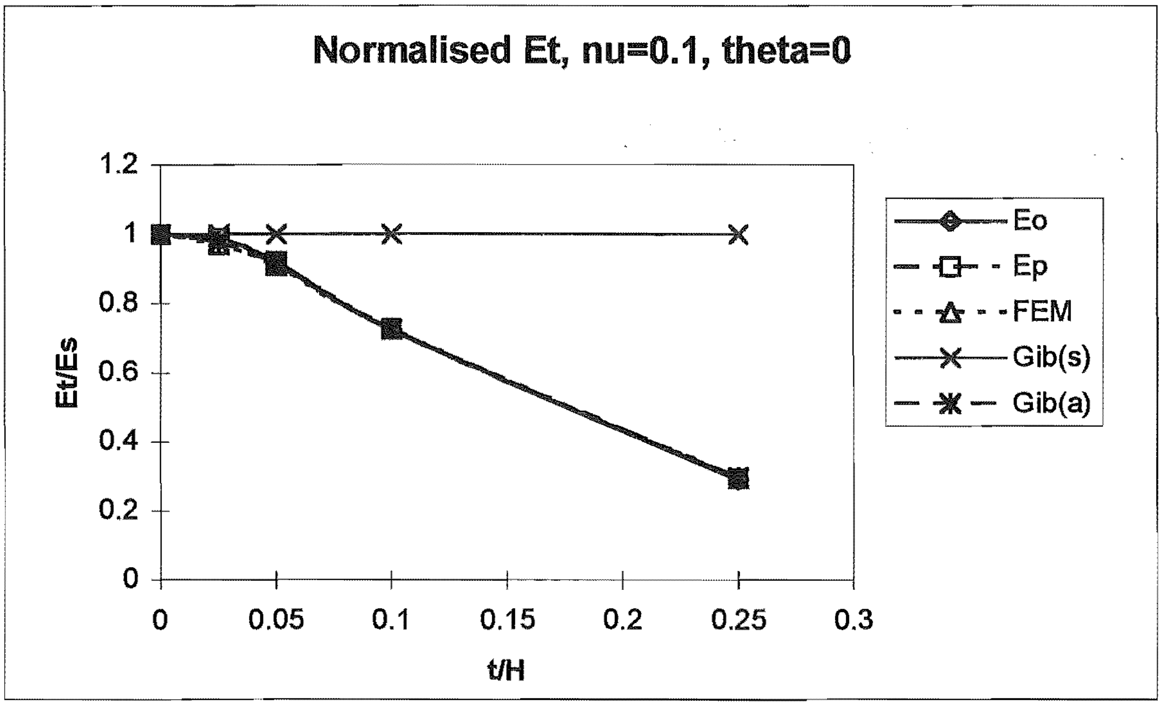


Figure 6.3 Et for cell angle =0, v=0.1

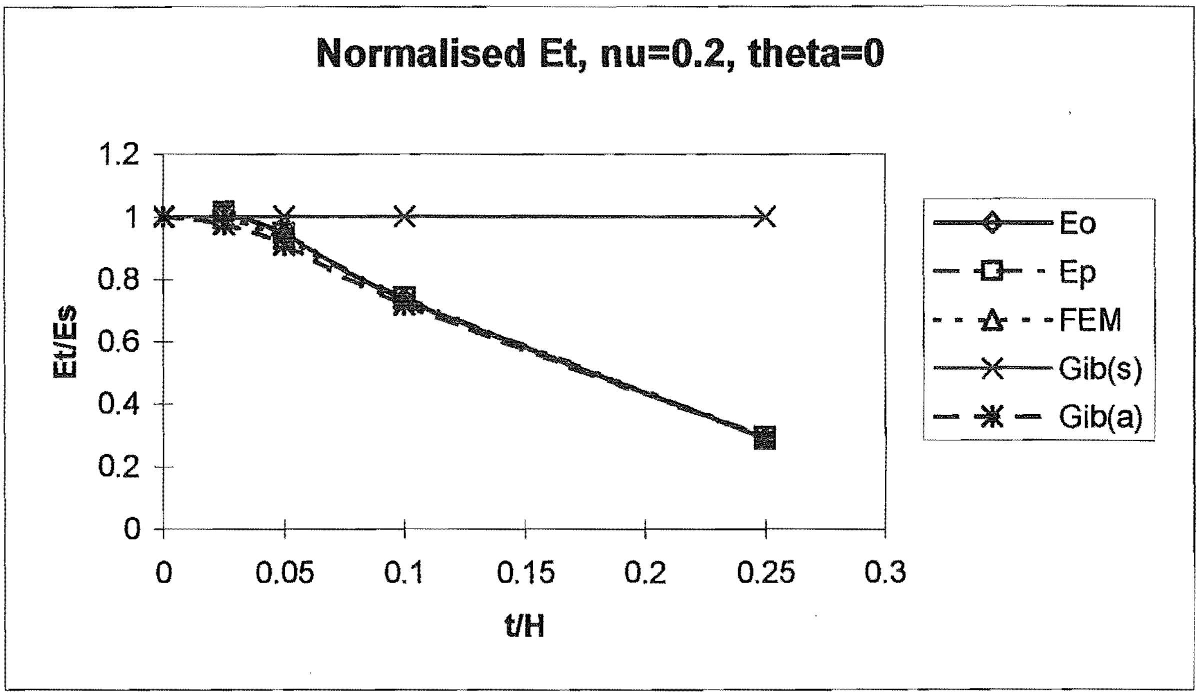


Figure 6.4 Et for cell angle=0, v=0.

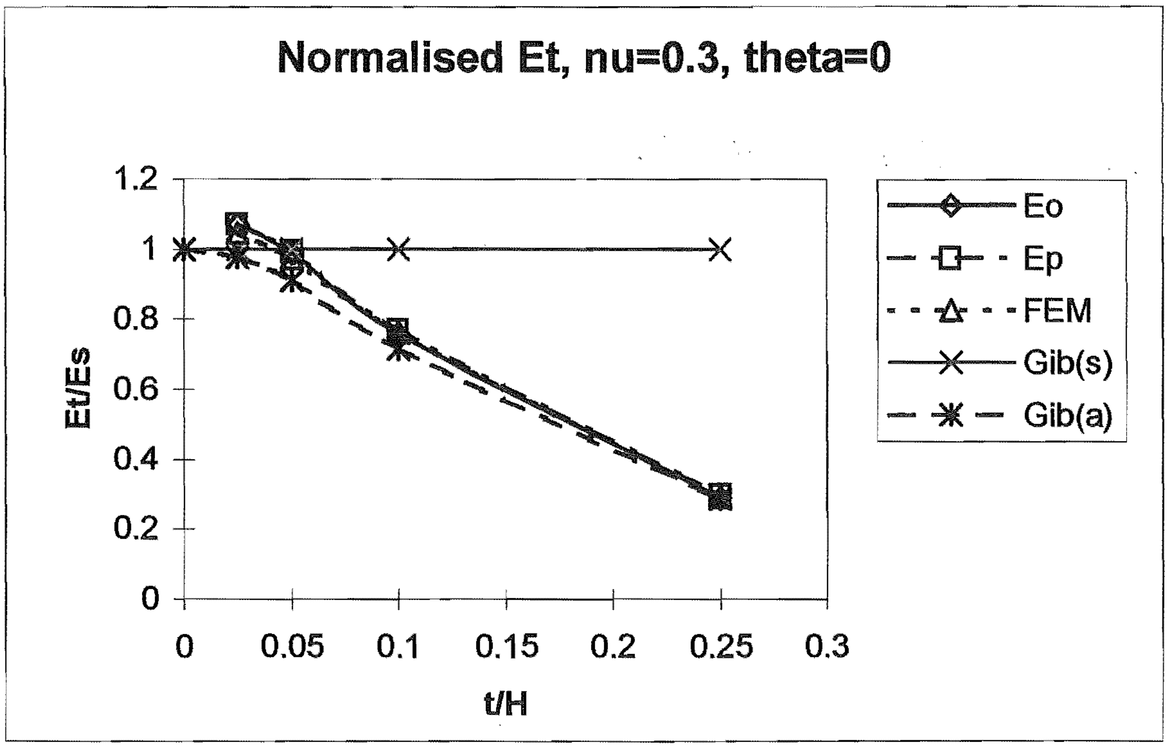


Figure 6.5 Et for cell angle=0, v=0.3.

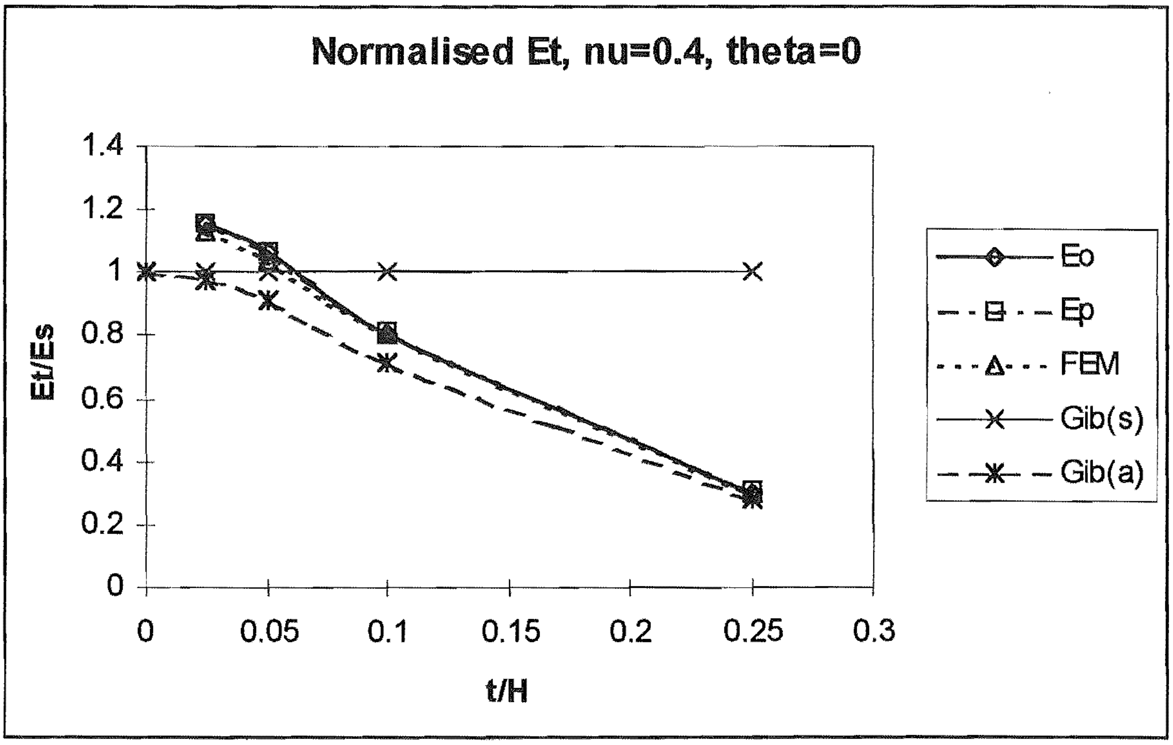
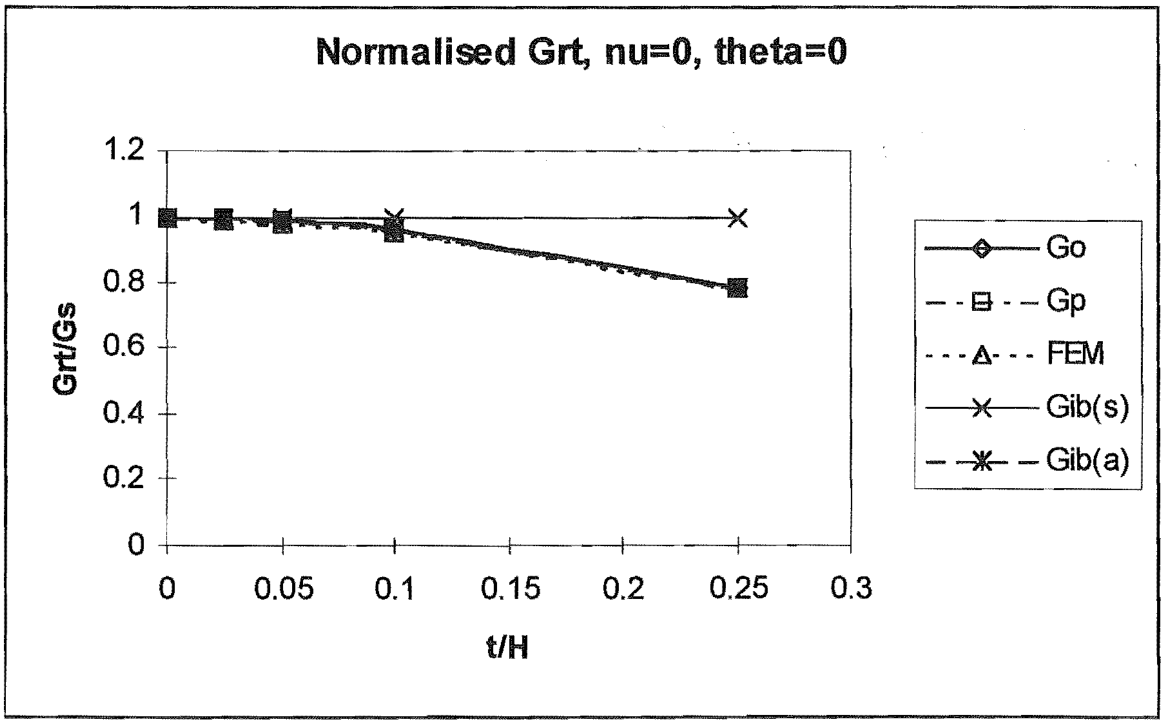


Figure 6.6 Et for cell angle =0,



Igure 6.7 Grt for cell angle=0, v=0

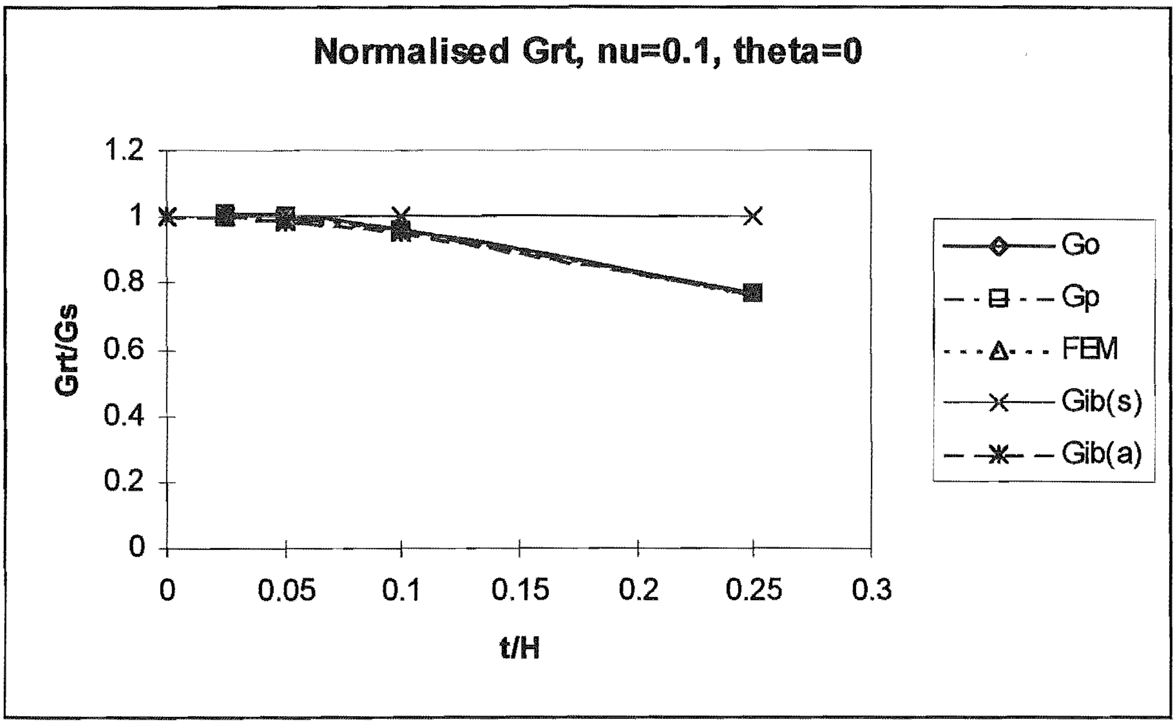


Figure 6.8 Grt for cell angle=0, v=0.1

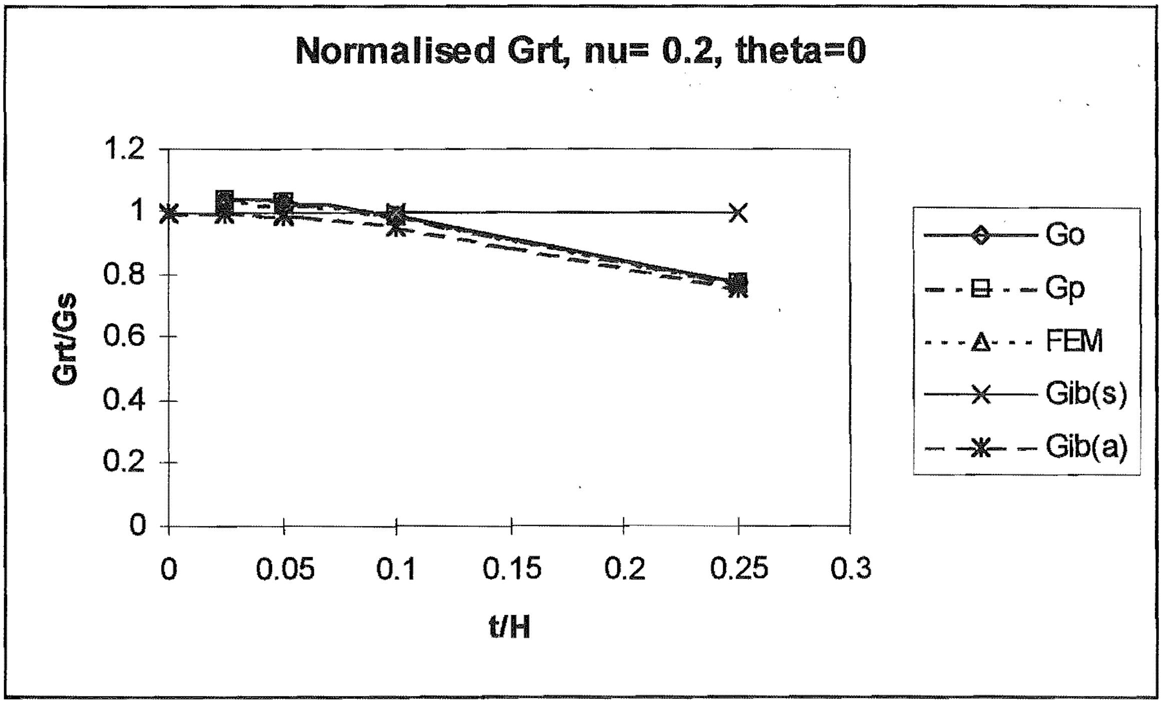


figure 6.9 Grt for cell angle=0, v=0.2

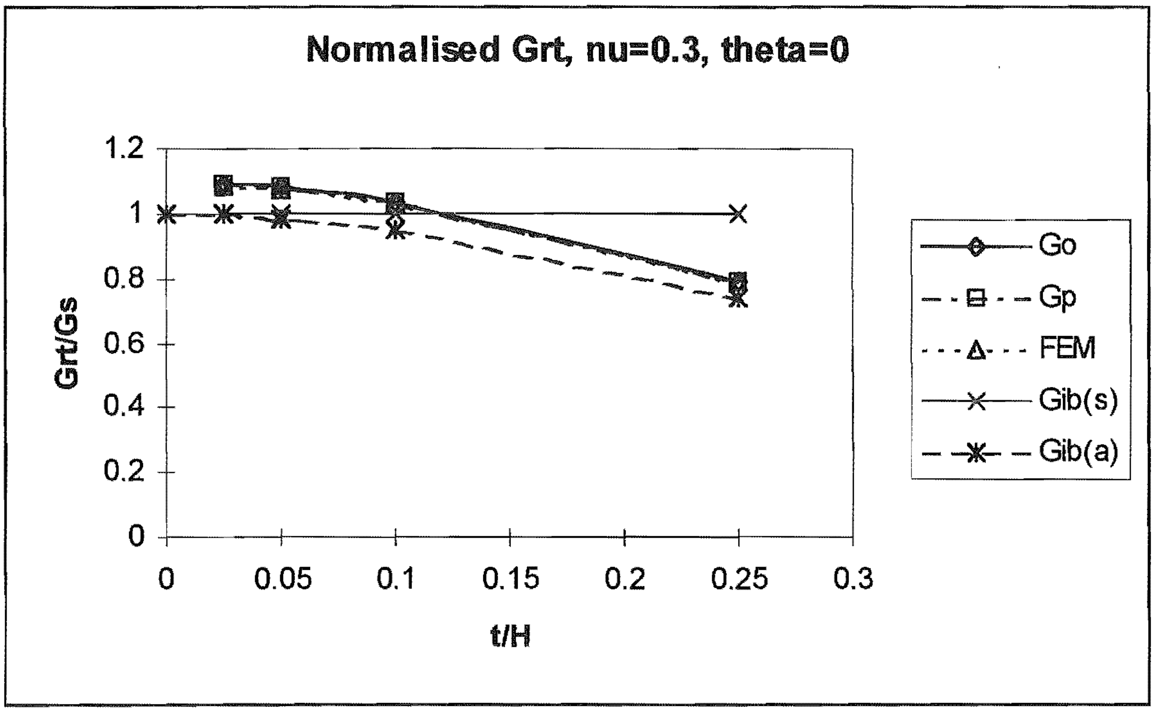
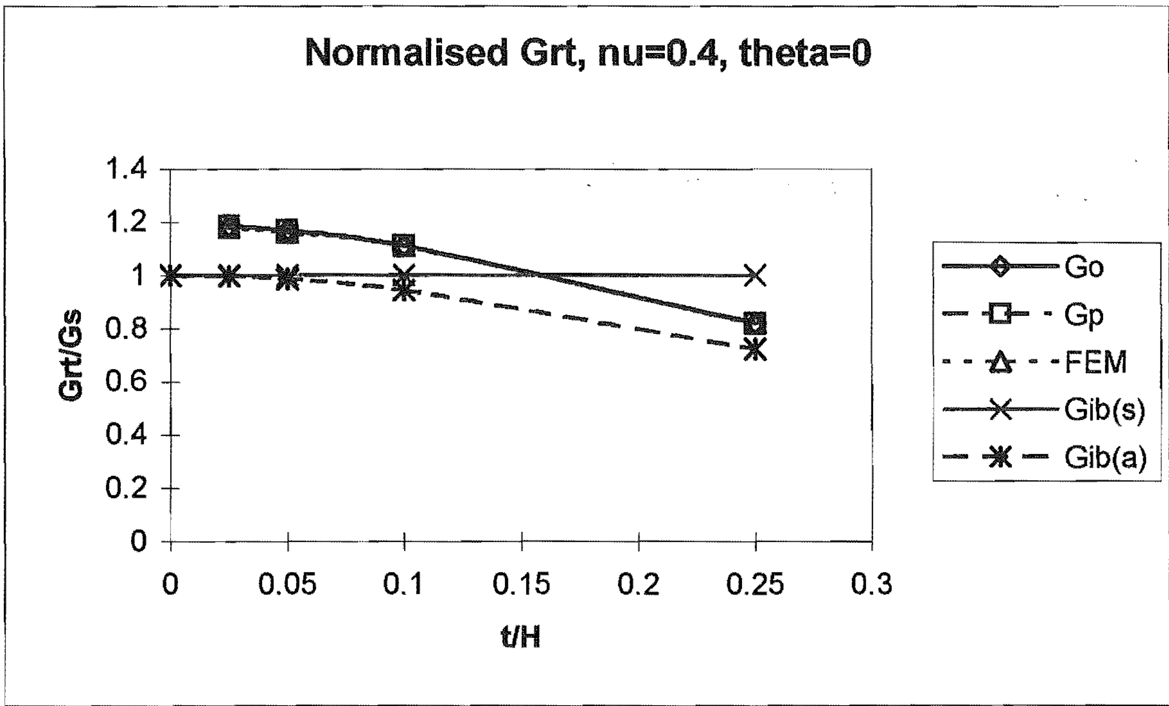


Figure 6.10 Grt for cell angle=0, v=0.3



Ogure 6.11 Grt for cell angle=0, v=0.4.

6.2 Graphs for analytical solutions, cell angle=15

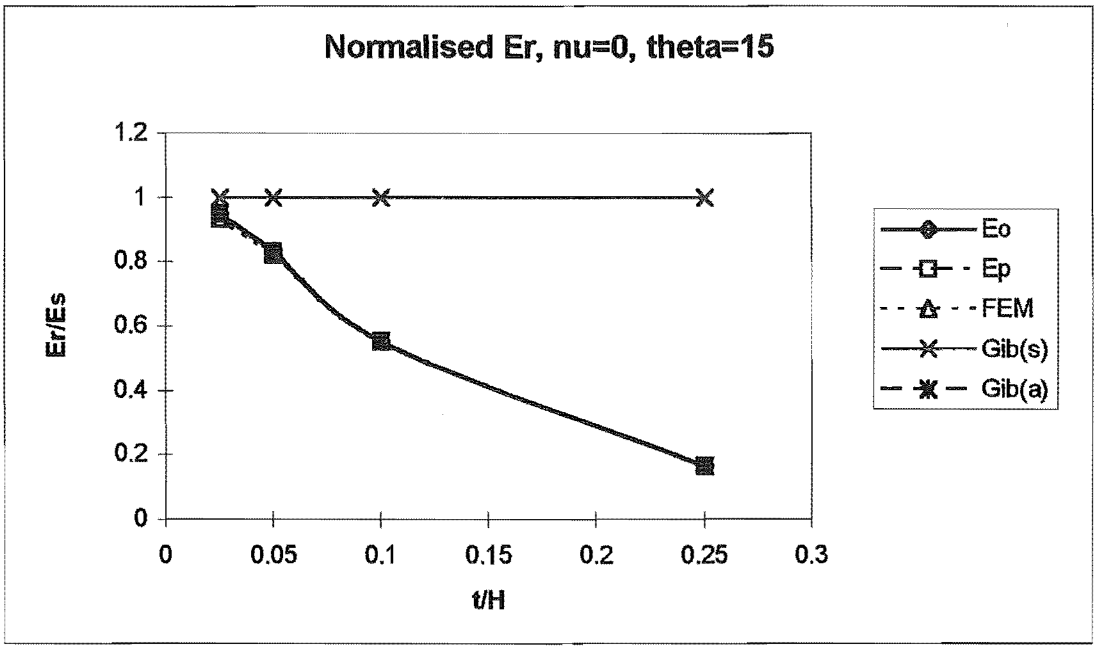


Figure 6.12 Er for cell angle=15, v=0

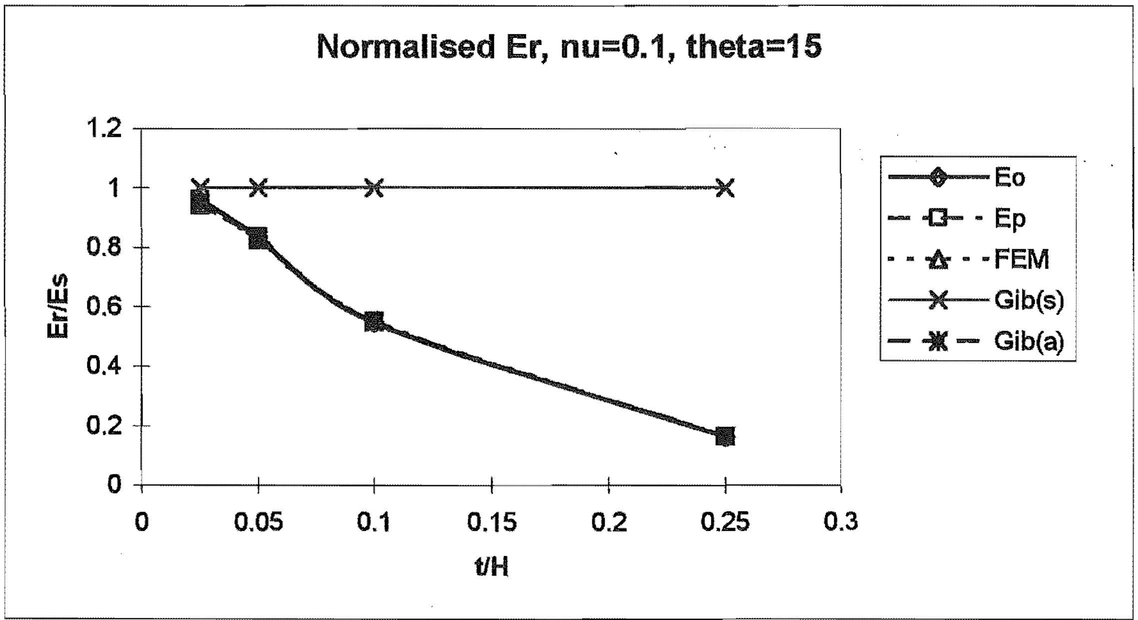


Figure 6.13 Er for cell angle=15, v=0.1

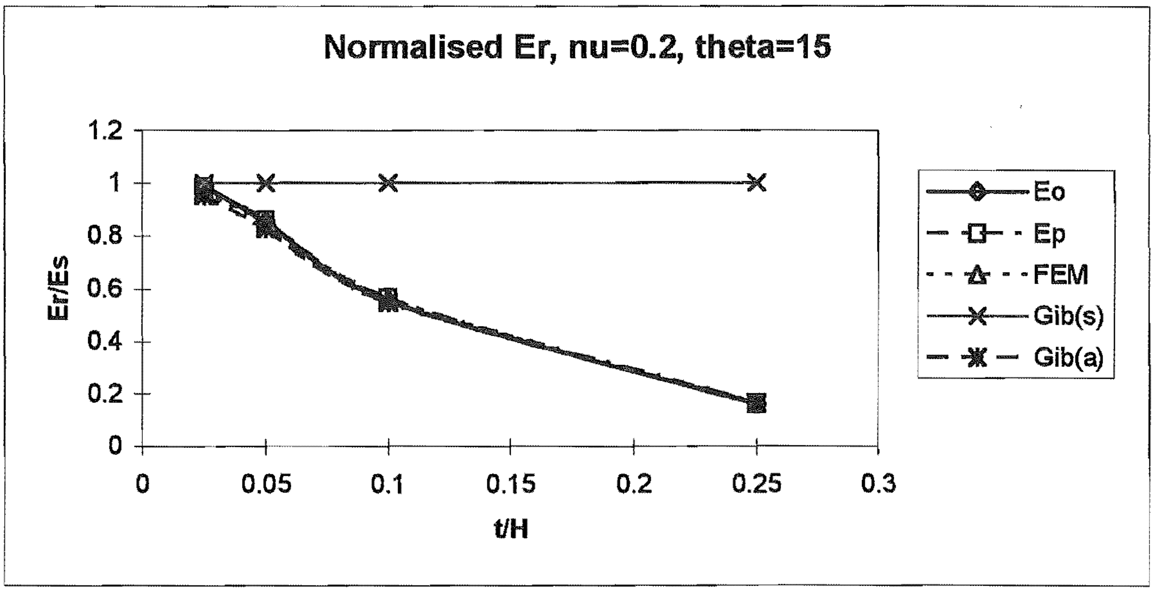


Figure 6.14 Er for cell angle=15, v=0.2

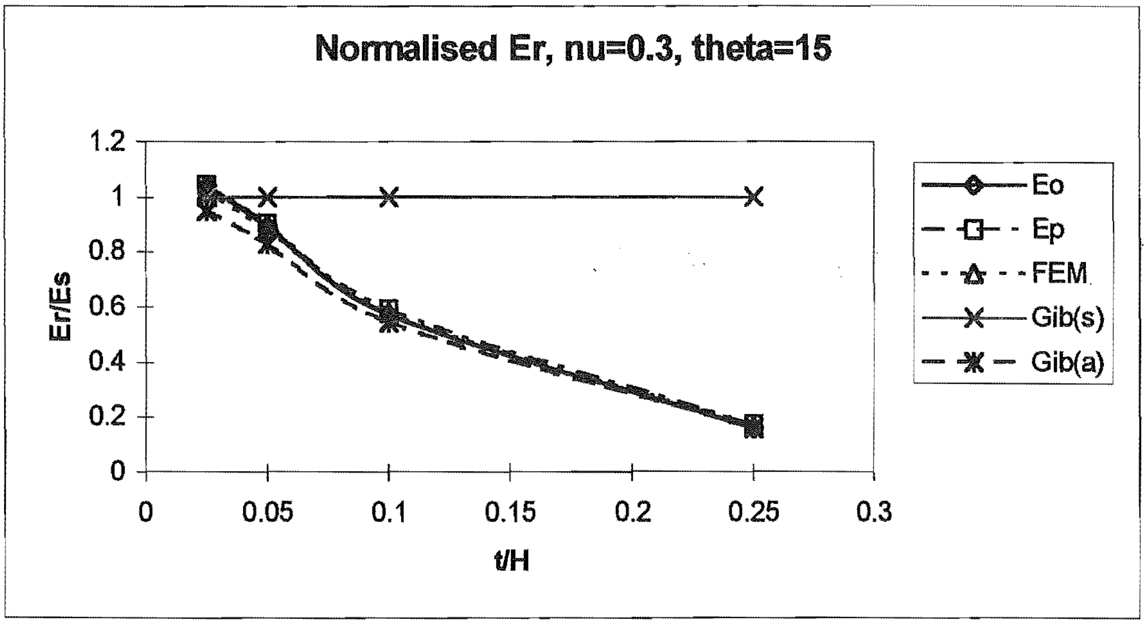


Figure 6.15 Er for cell angle=15, v=0.3

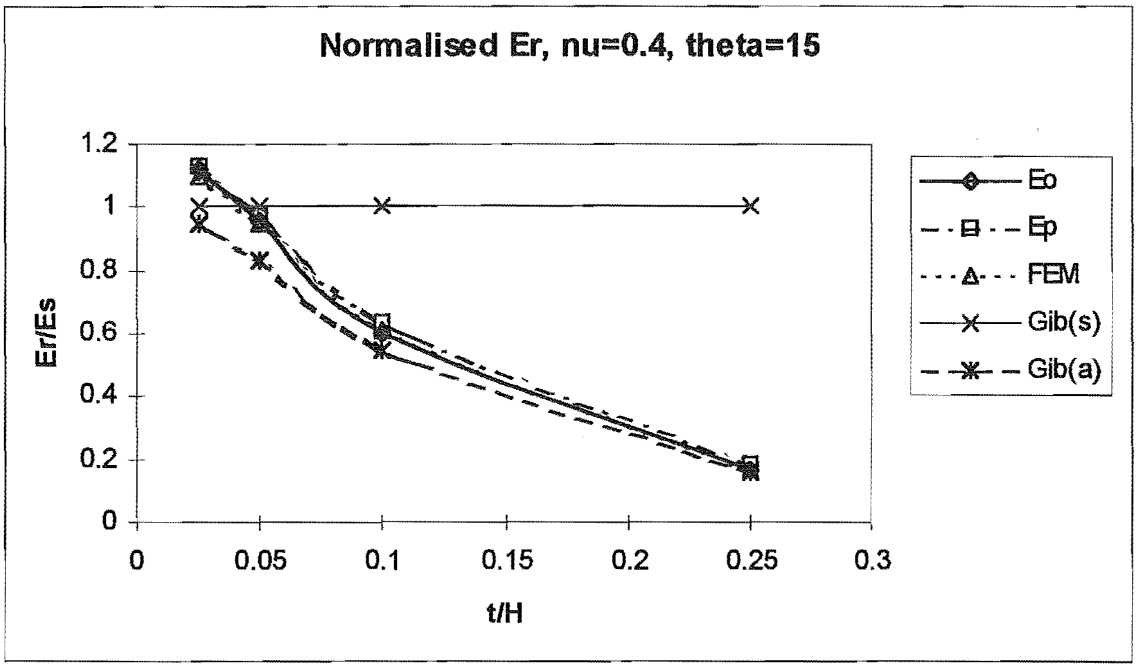


Figure 6.16 Er for cell angle=15, v=0.4

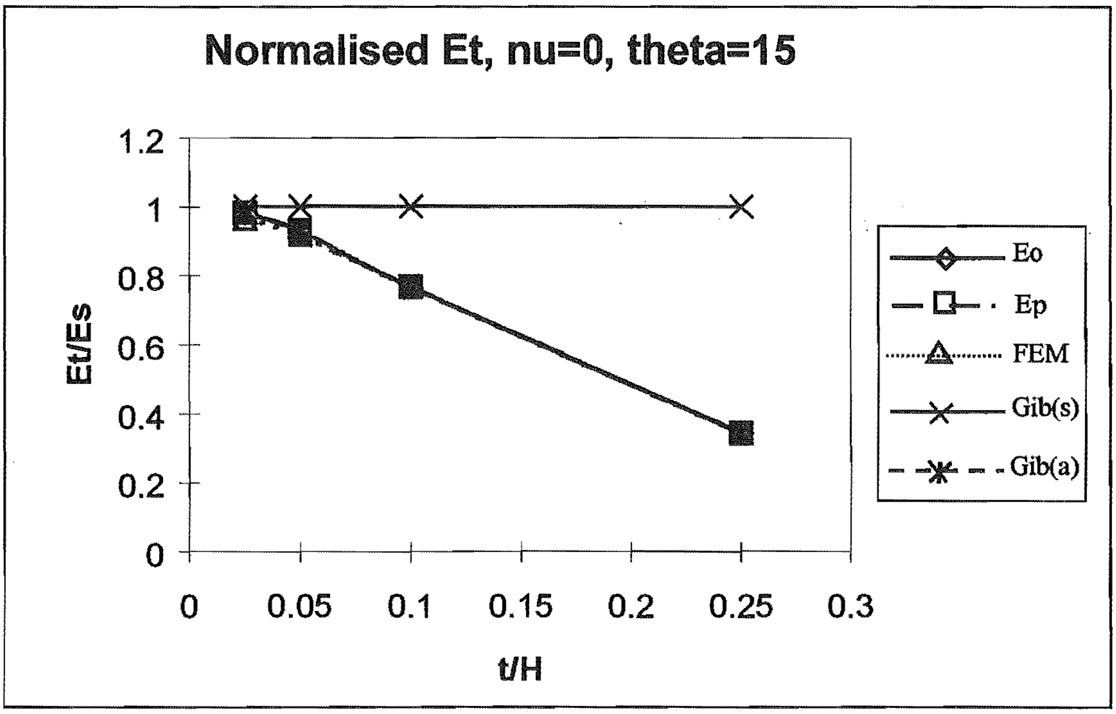


Figure 6.17 Et for cell angle=15, v=0

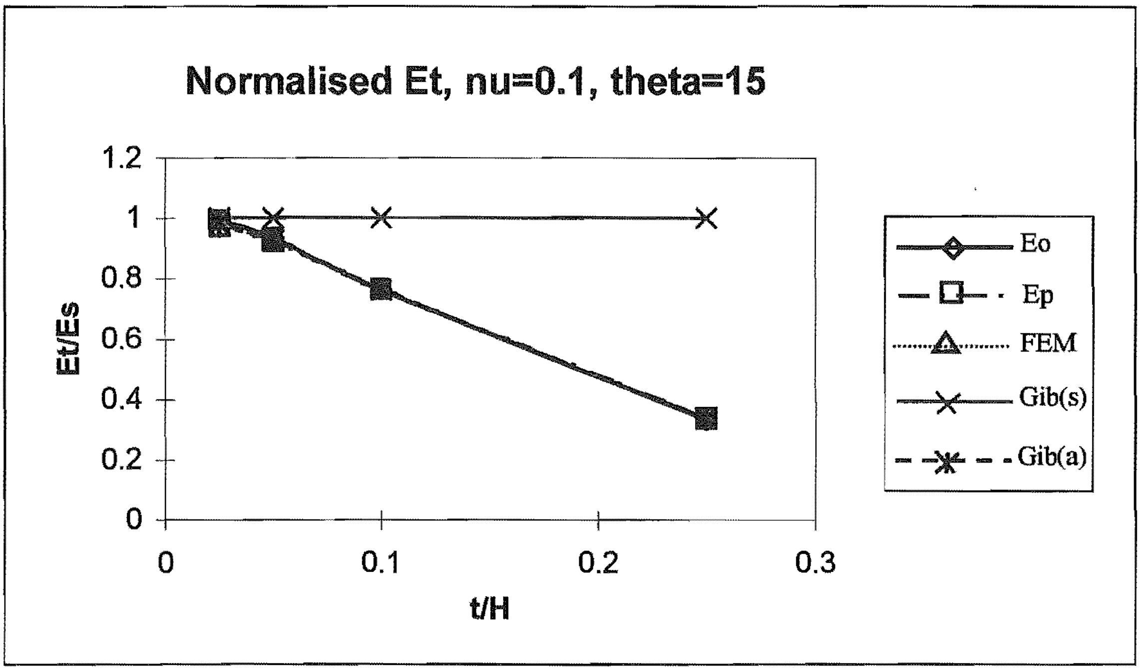


Figure 6.18 Et for cell angle=15, v=0.1

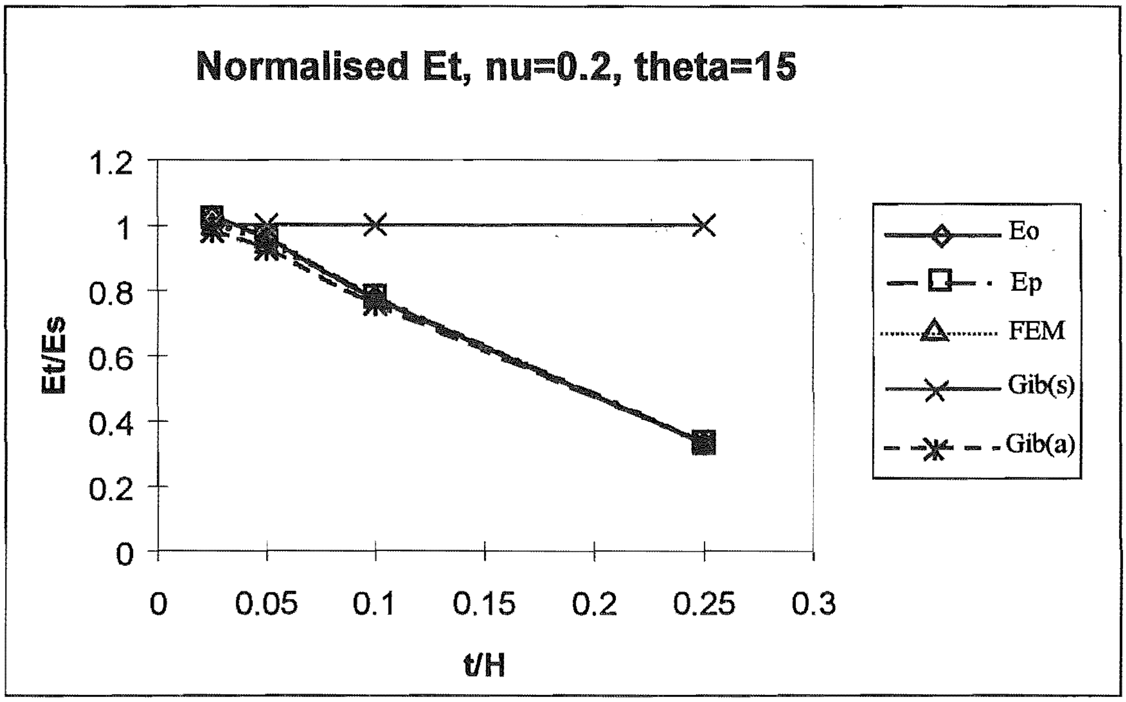


Figure 6.19 Et for cell angle=15, v=0.2

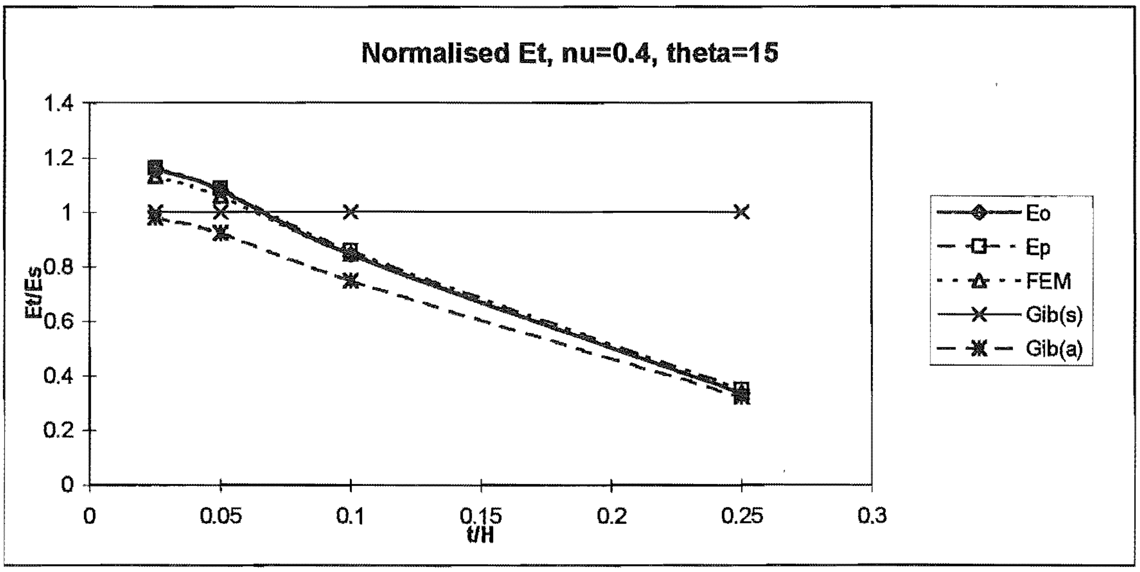


Figure 6.20 Et for cell angle=14, v=0.4

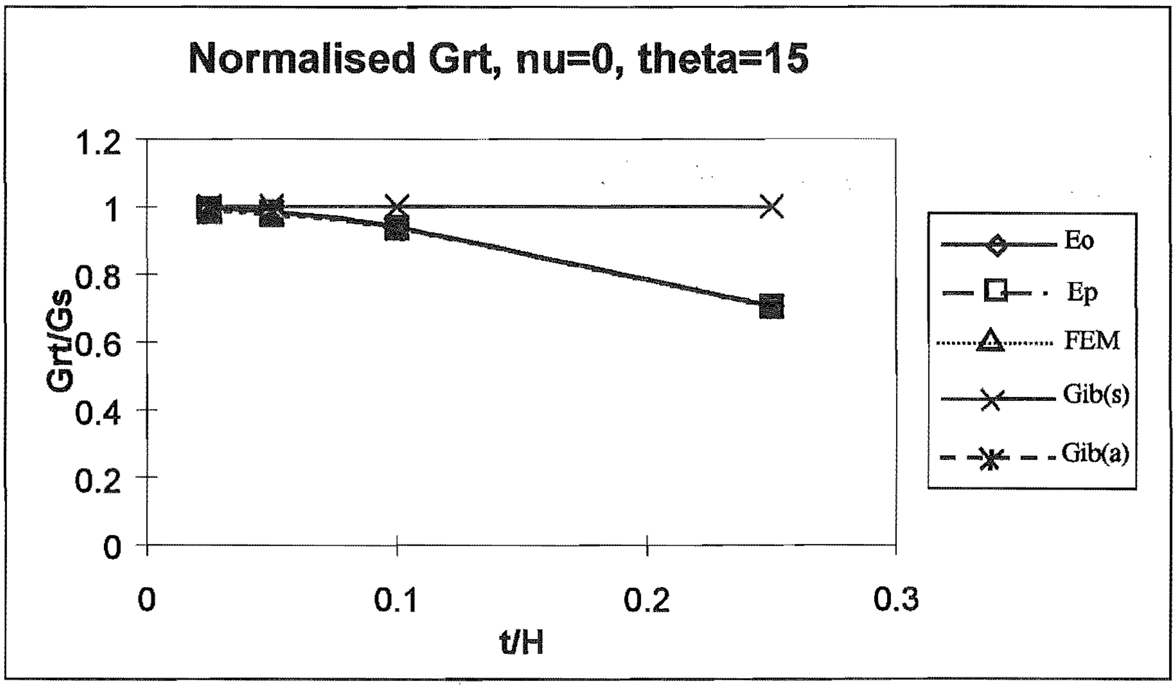


Figure 6.21 Grt for cell angle=15, v=0

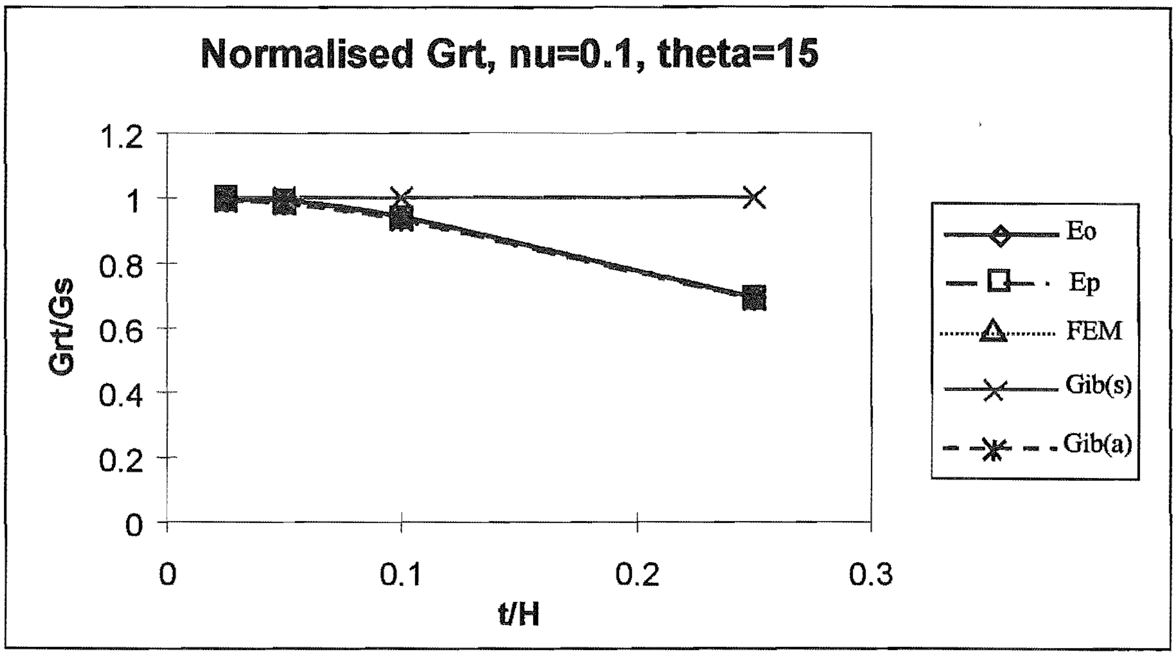


Figure 6.22 Grt for cell angle=15, v=0.1

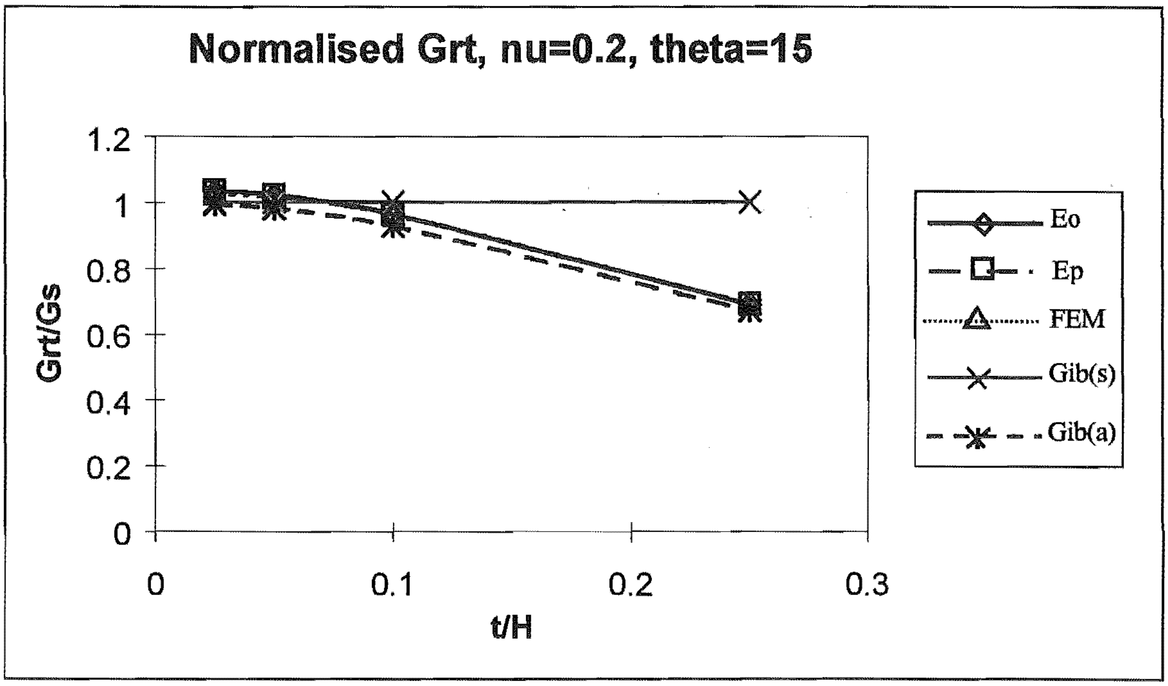


Figure 6.23 Grt for cell angle=15, v=0.2

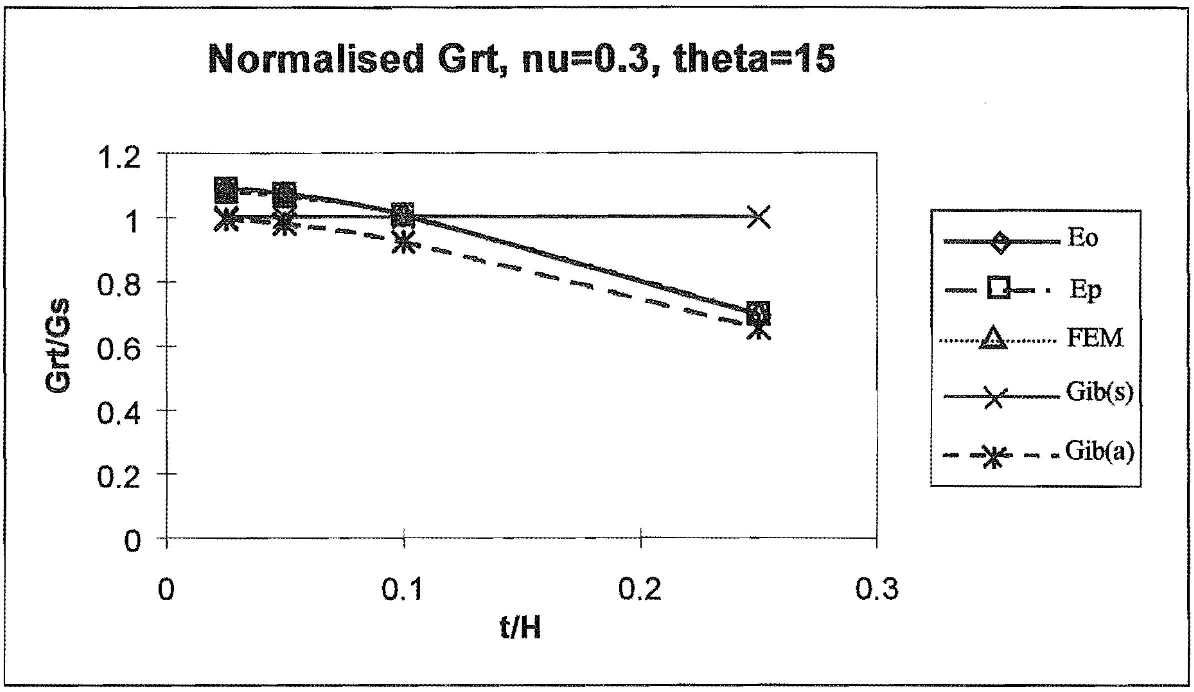
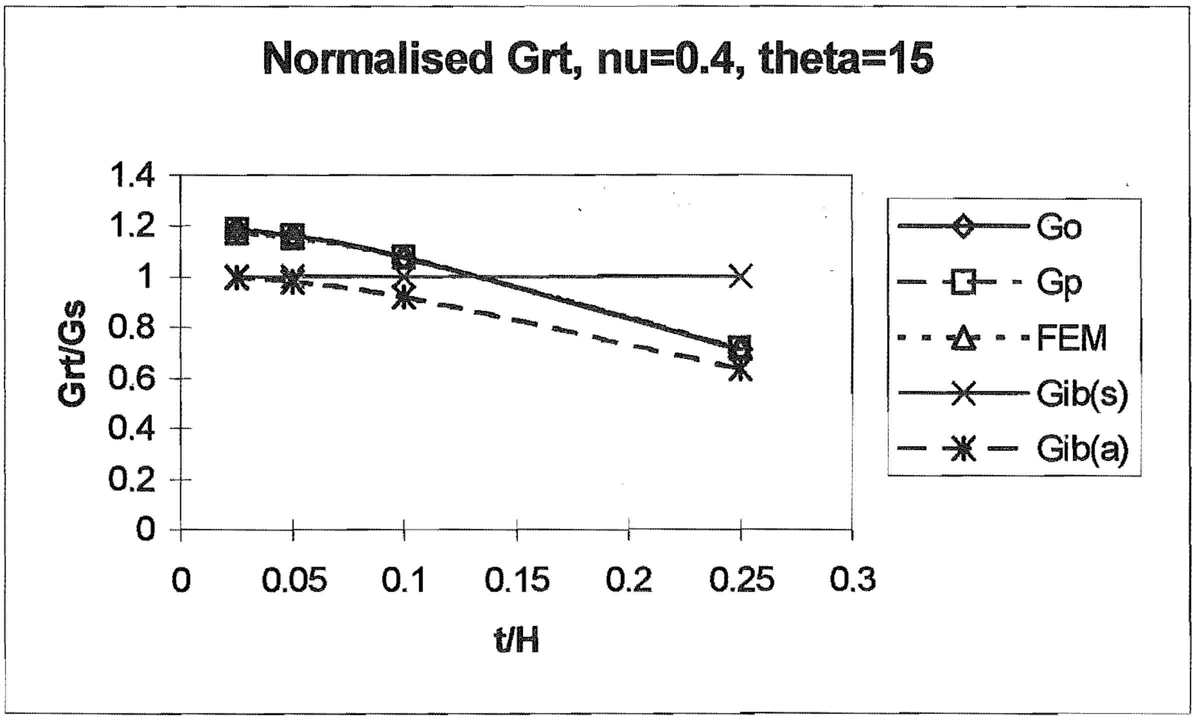


Figure 6.24 Grt for cell angle=15, v=0.3



Figre 6.25 Grt for cell angle=15, v=0.4

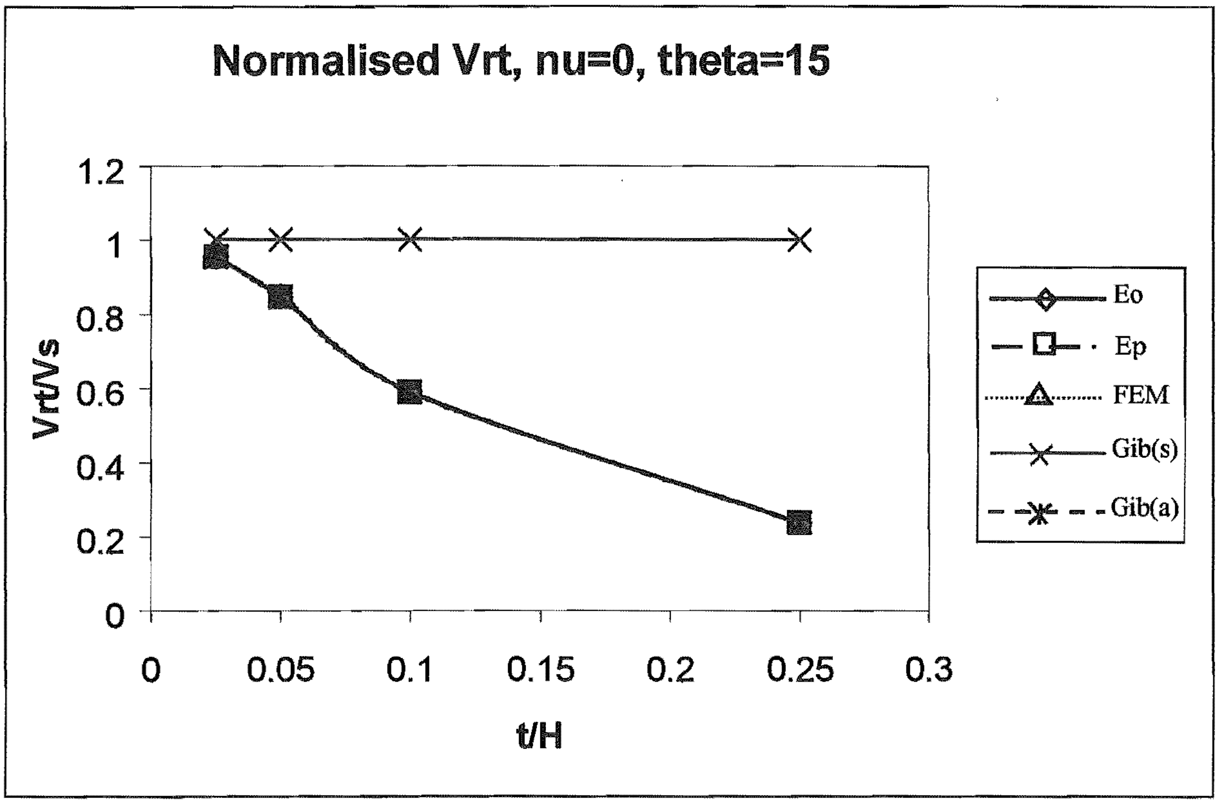


Figure 6.26 Vrt for cell angle=15, v=0

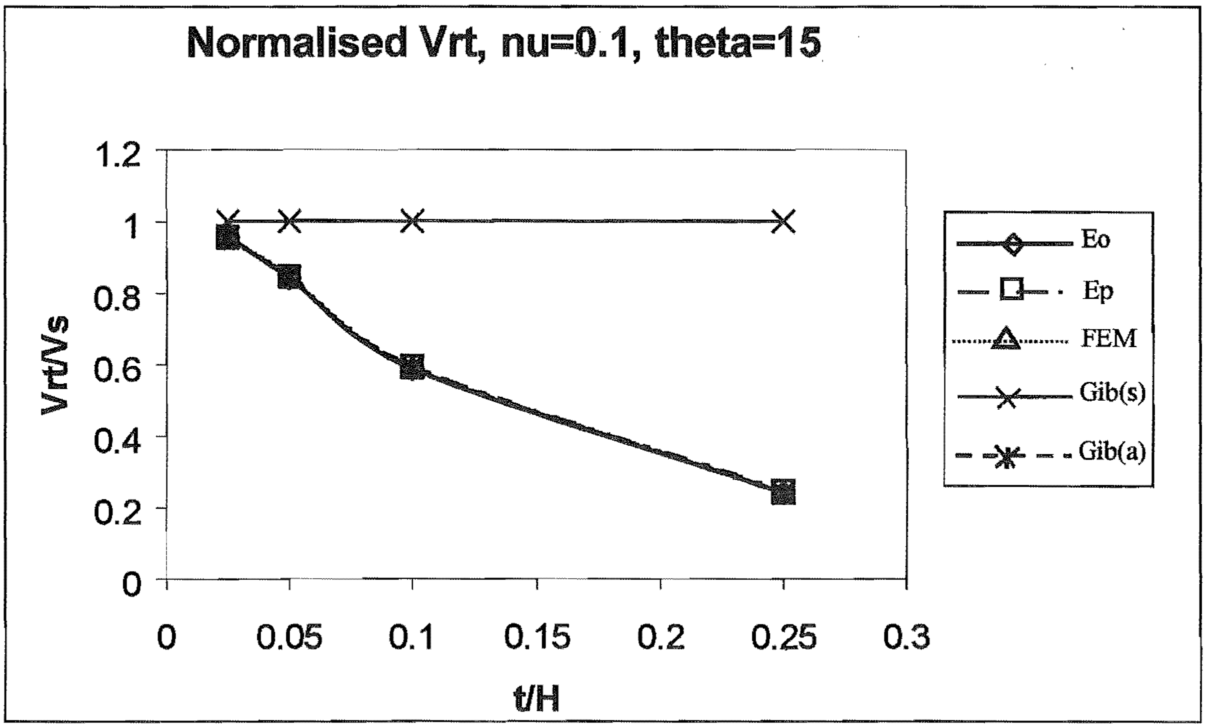


Figure 6.27 Vrt for cell angle=15, v=0.1

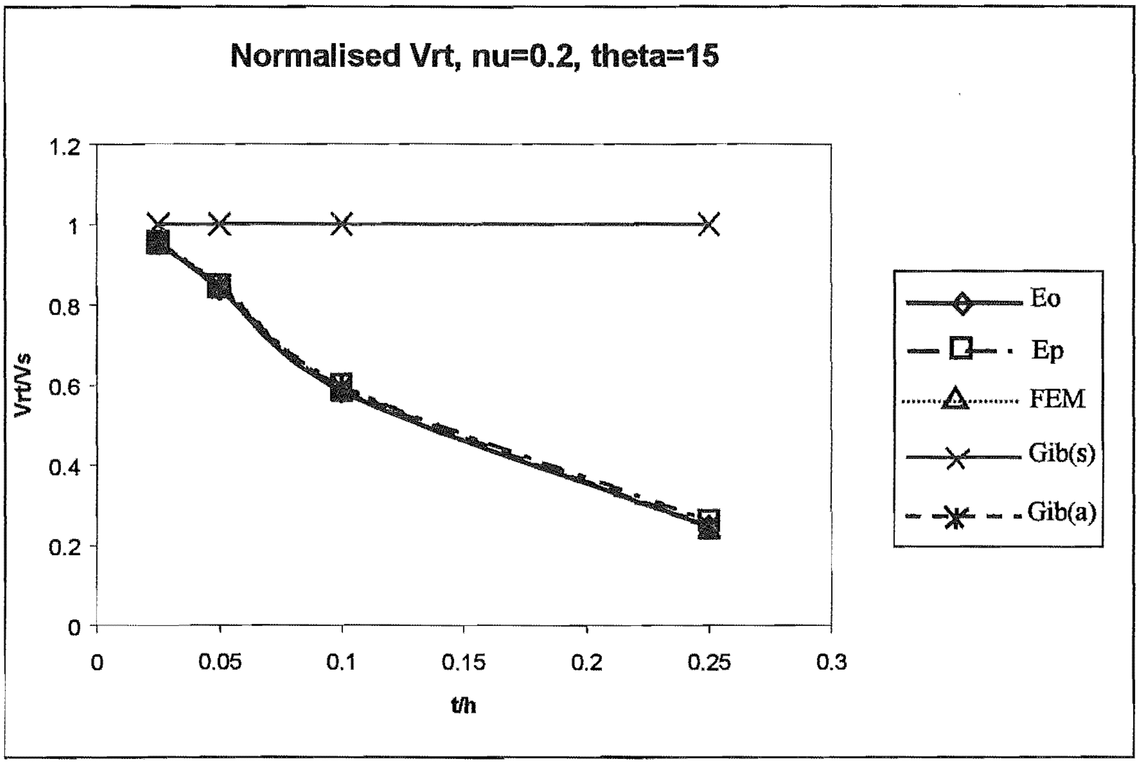


Figure 6. 28 Vrt for cell angle=15, v=0.2

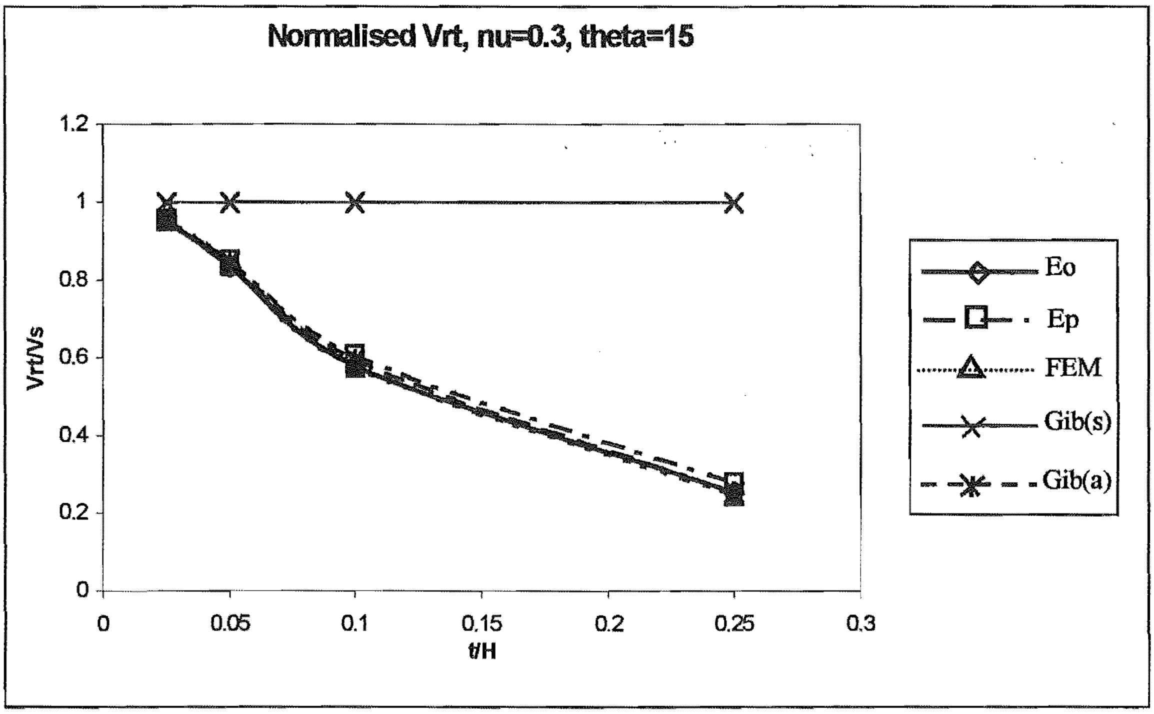


Figure 6.29 Vrt for cell angle=15, v=0.3

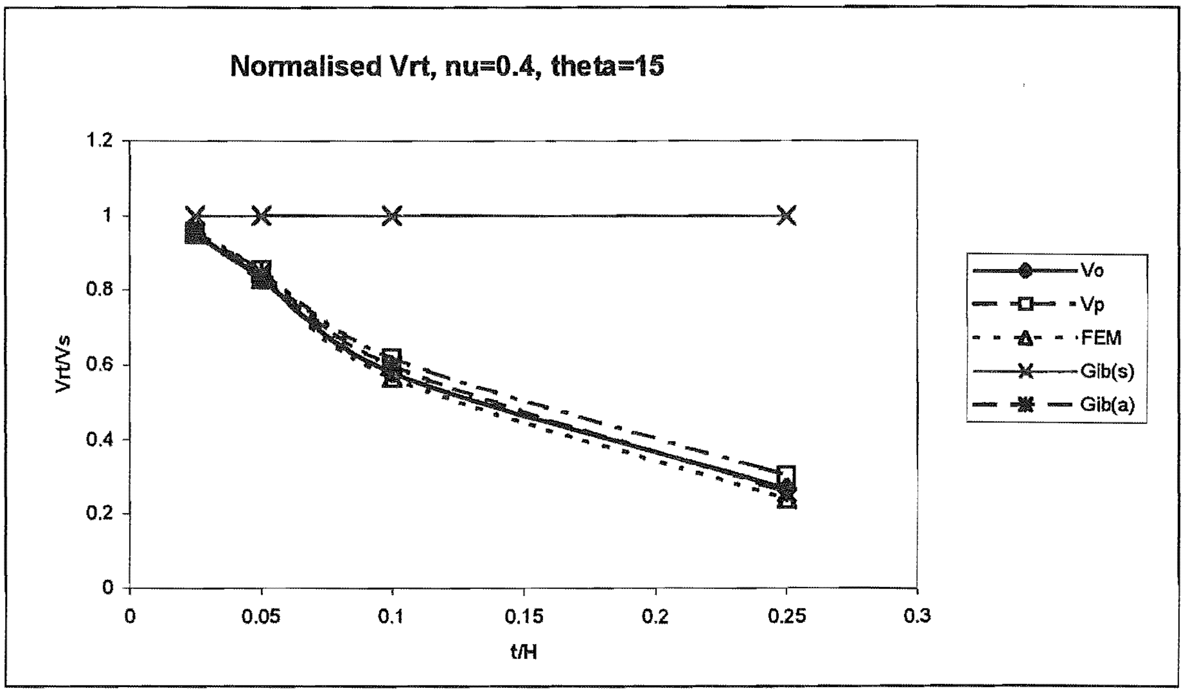


Figure 6.30 Vrt for cell angle=15, v=0.4

6.3 Graphs of analytical solutions, cell angle=30

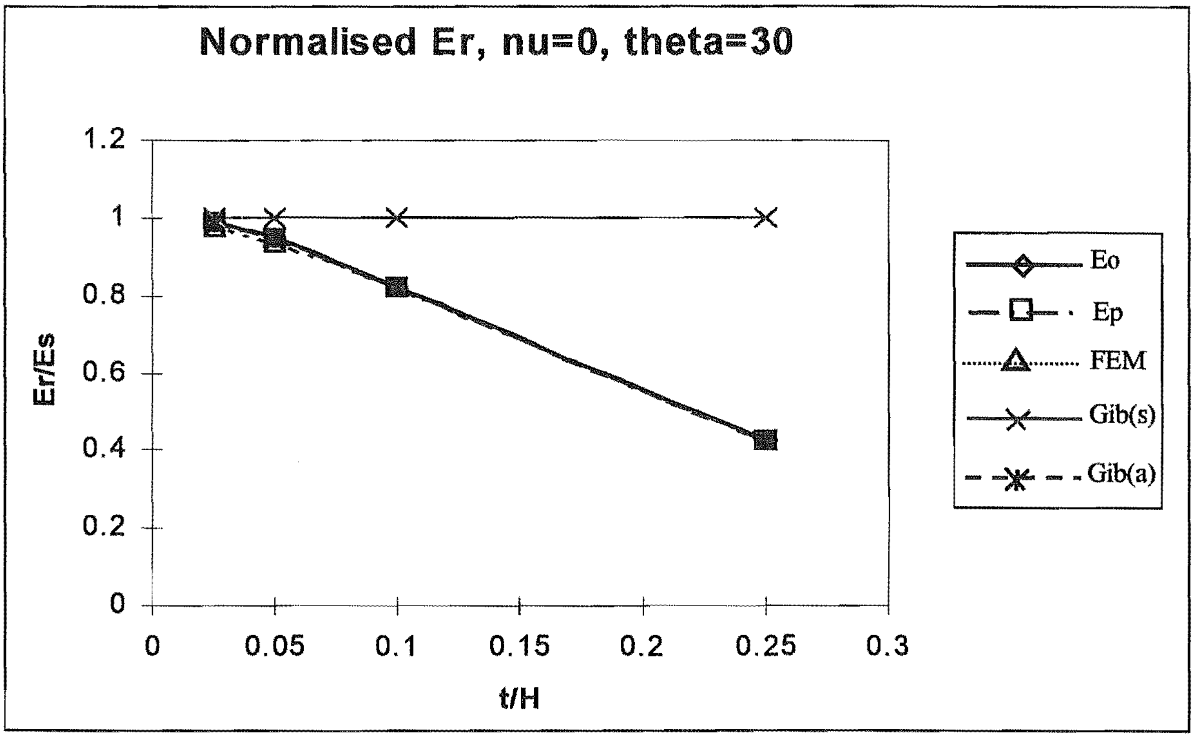


Figure 6.31 Er for cell angle=30, v=0

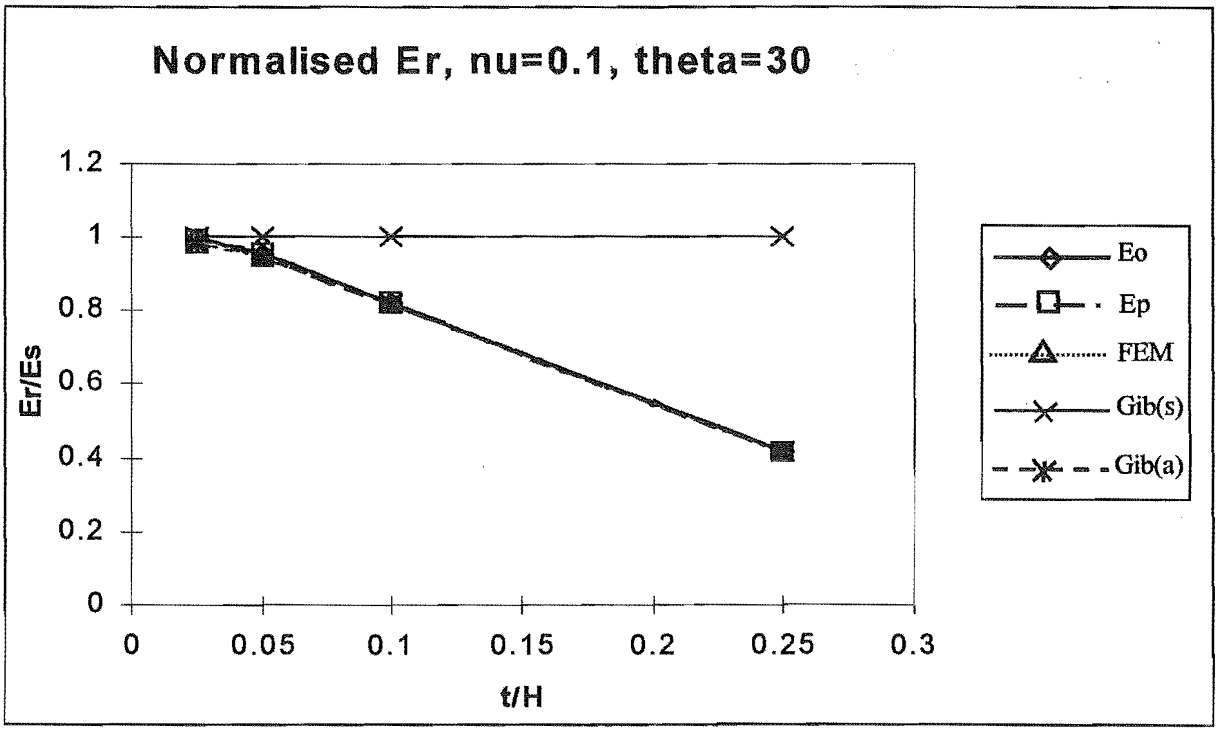


Figure 6.32 Er for cell angle=30, v=0.1

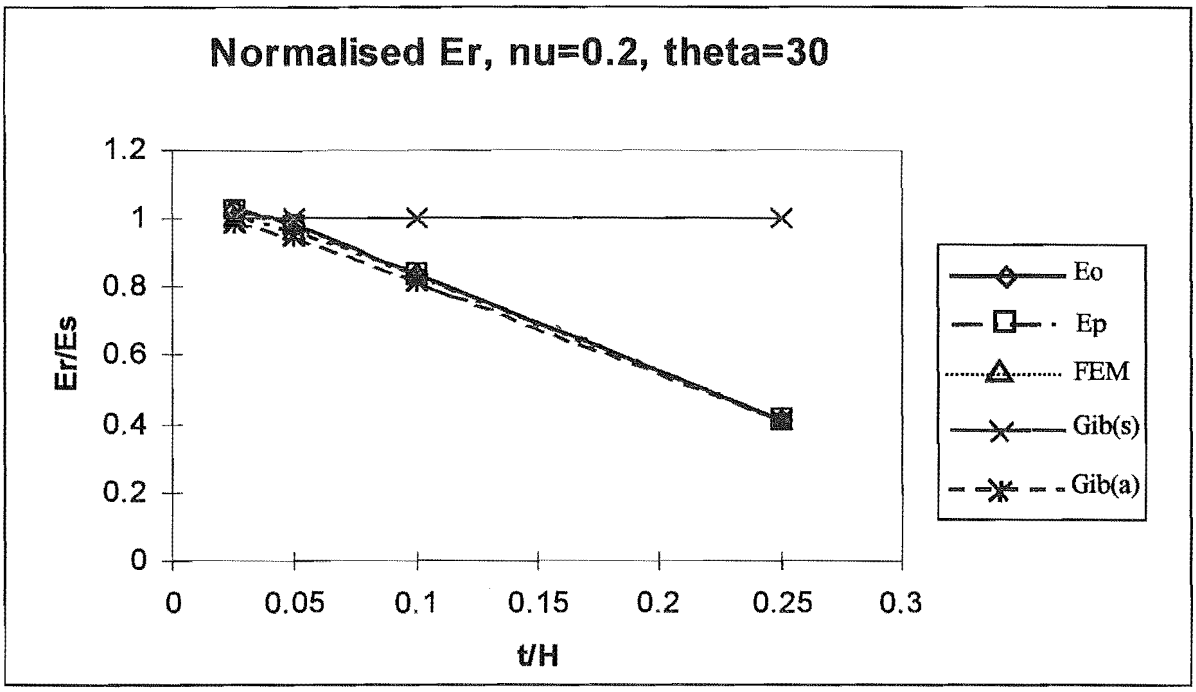


Figure 6.33 Er for cell angle=30, v=0.2

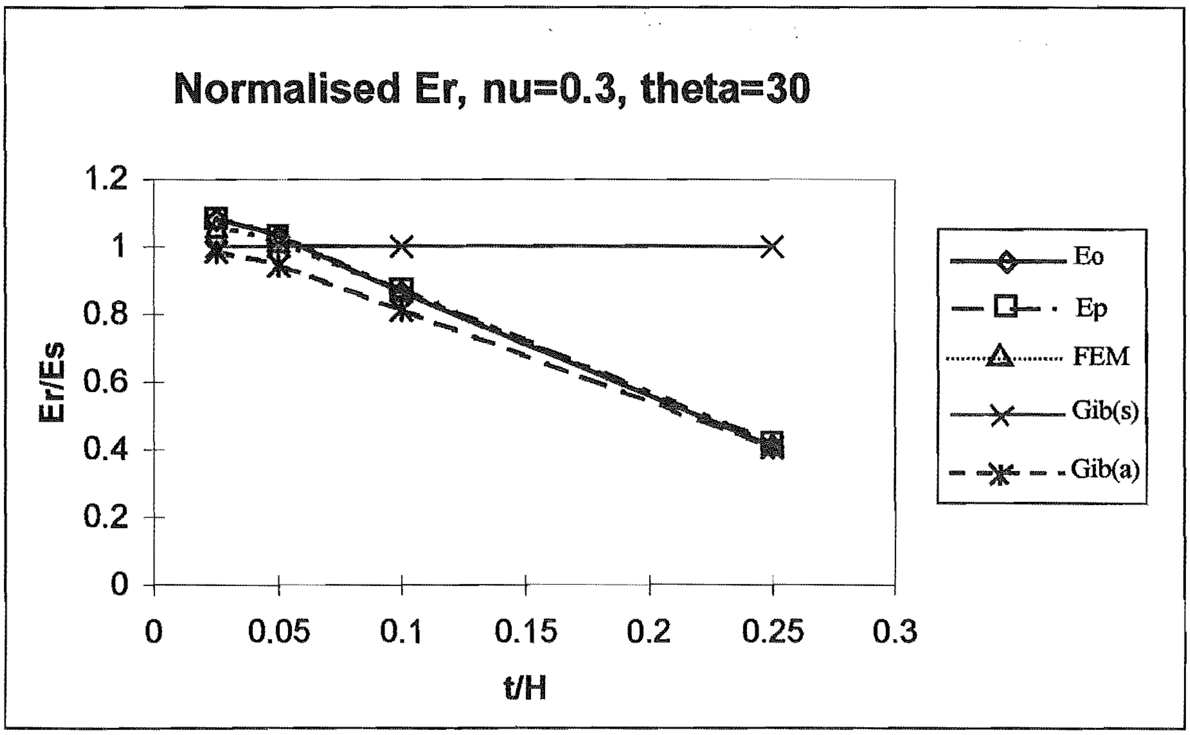


Figure 6.34 Er for cell angle=30, v=0.3

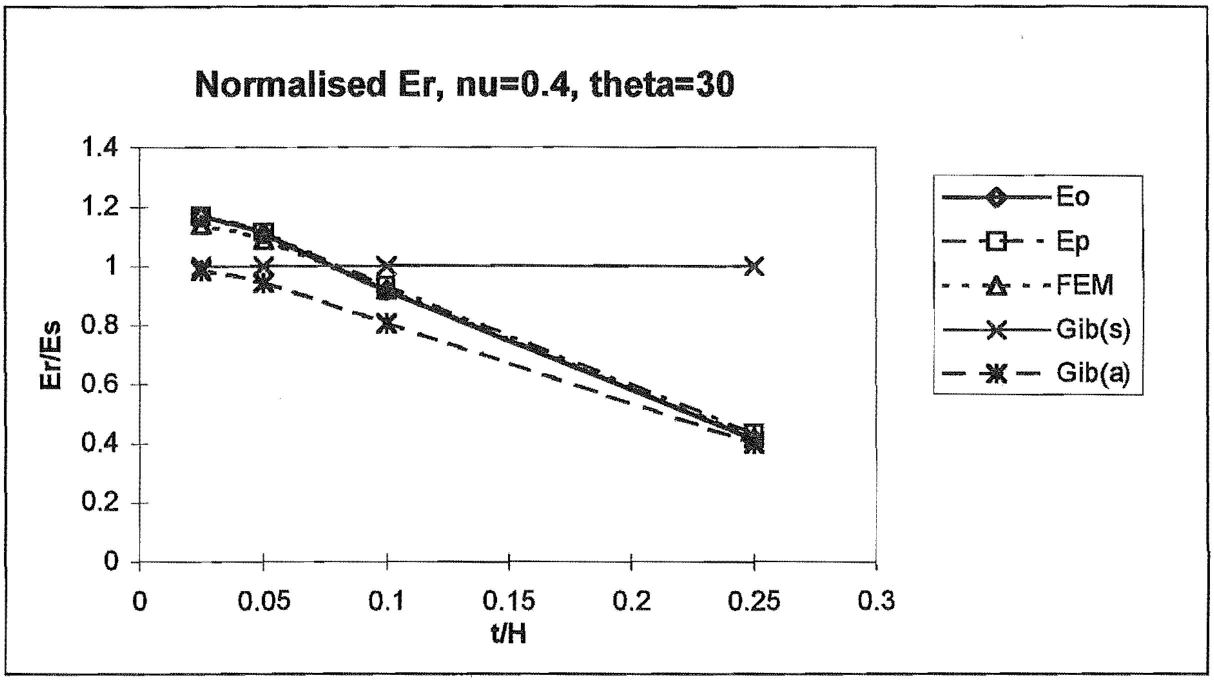


Figure 6.35 Er for cell angle=30, v=0.4

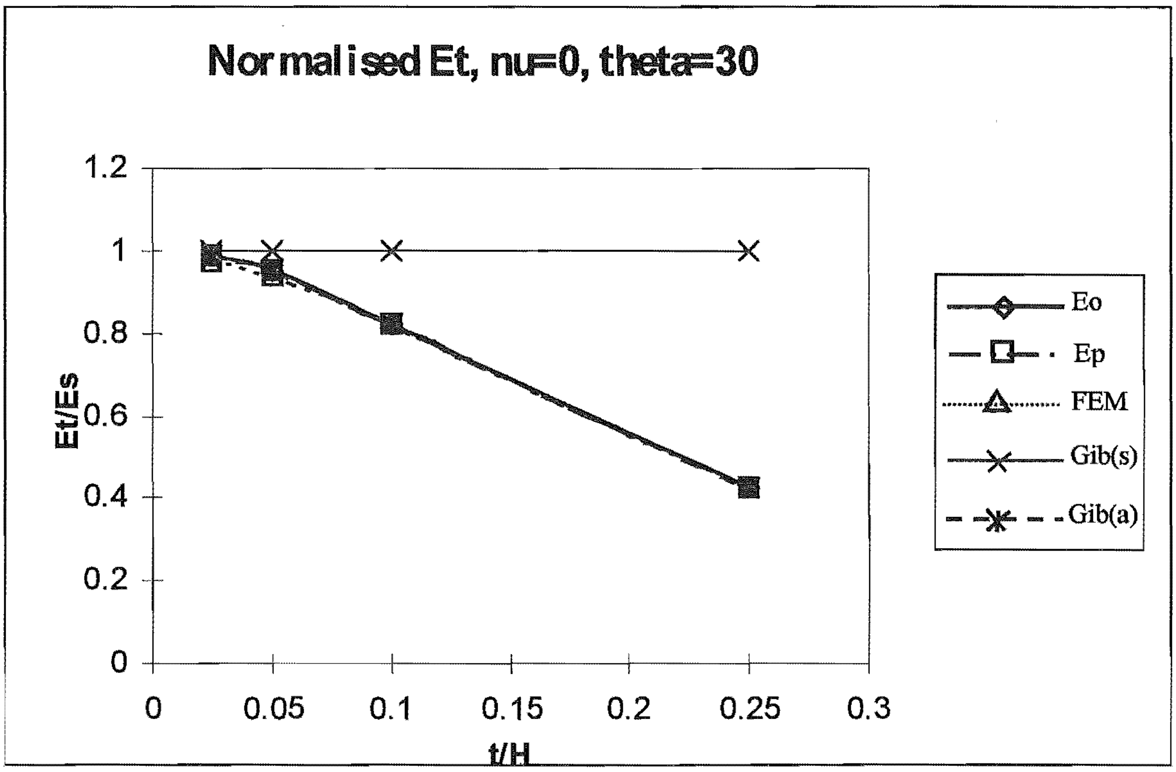


Figure 6.36 Et for cell angle=30, v=0

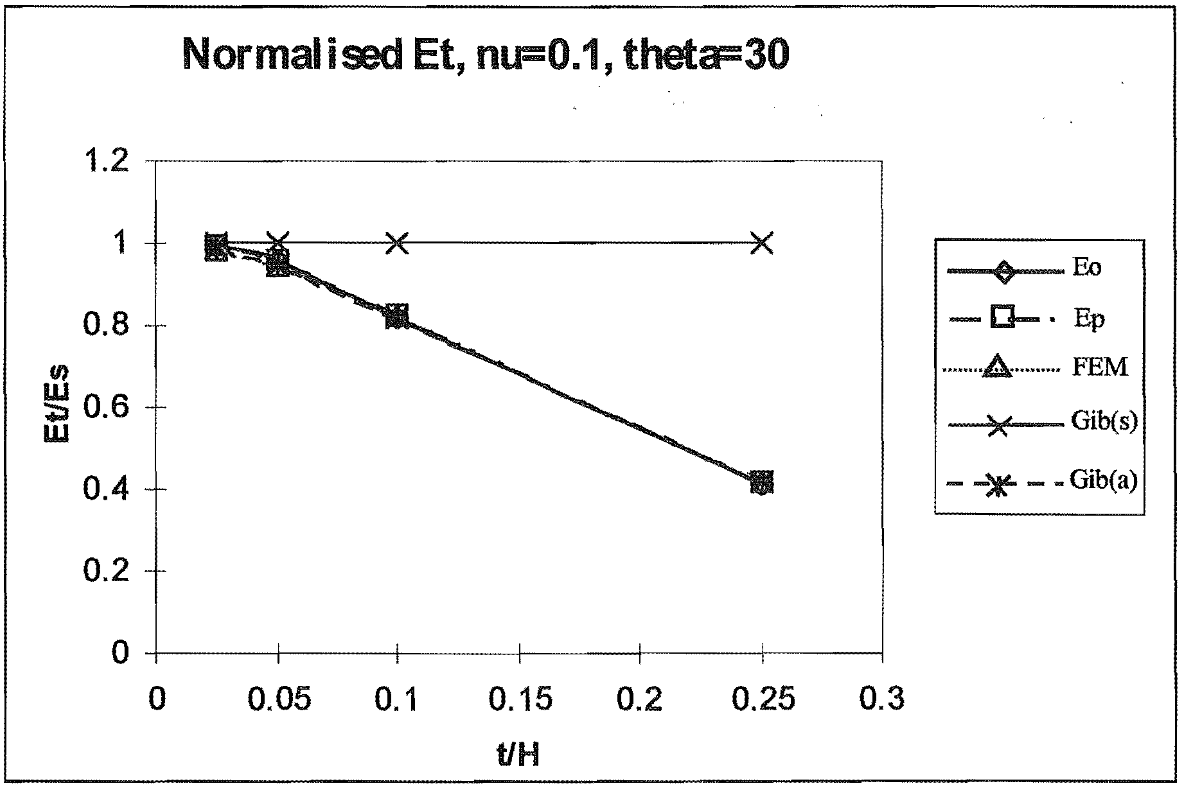


Figure 6.37 Et for cell angle=30, v=0.1

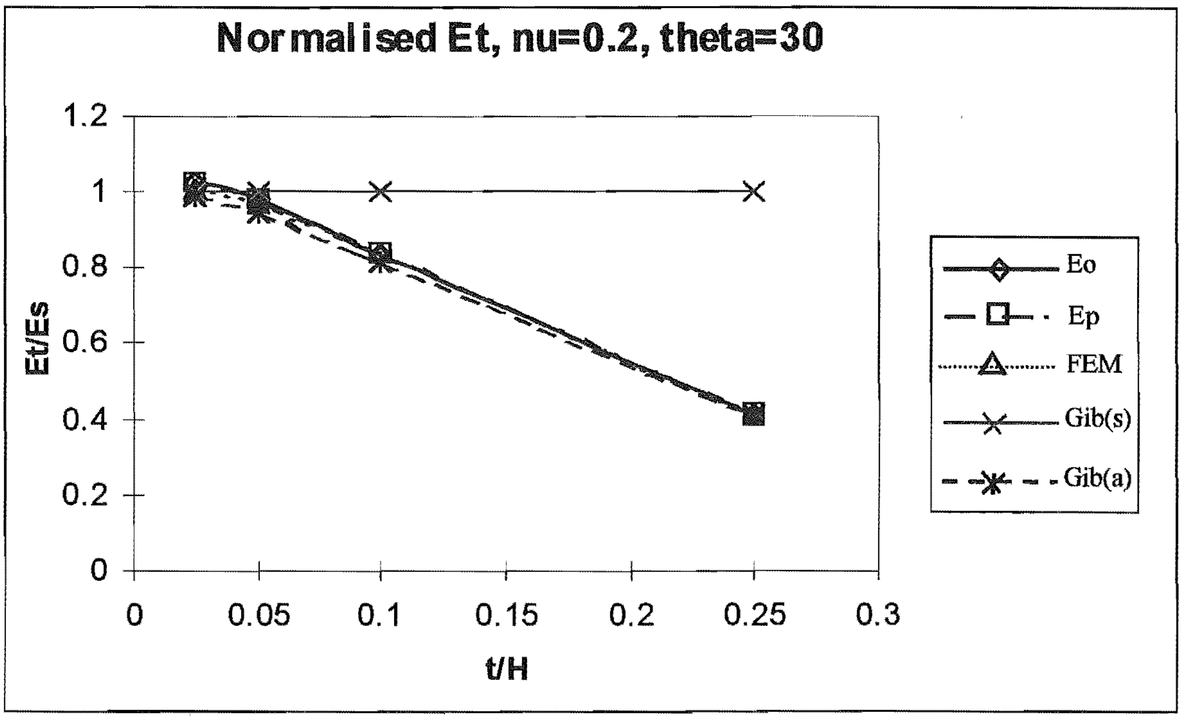


Figure 6.38 Et for cell angle=30, v=0.2

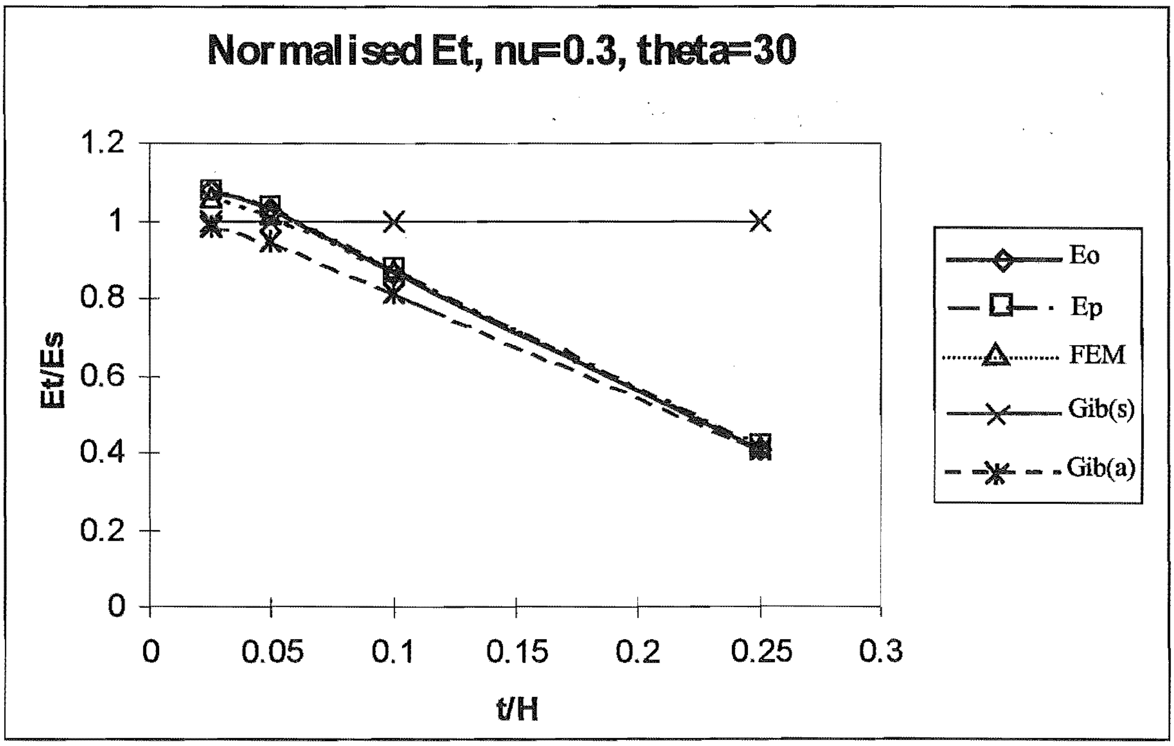


Figure 6.39 Et for cell angle=30, v=0.3

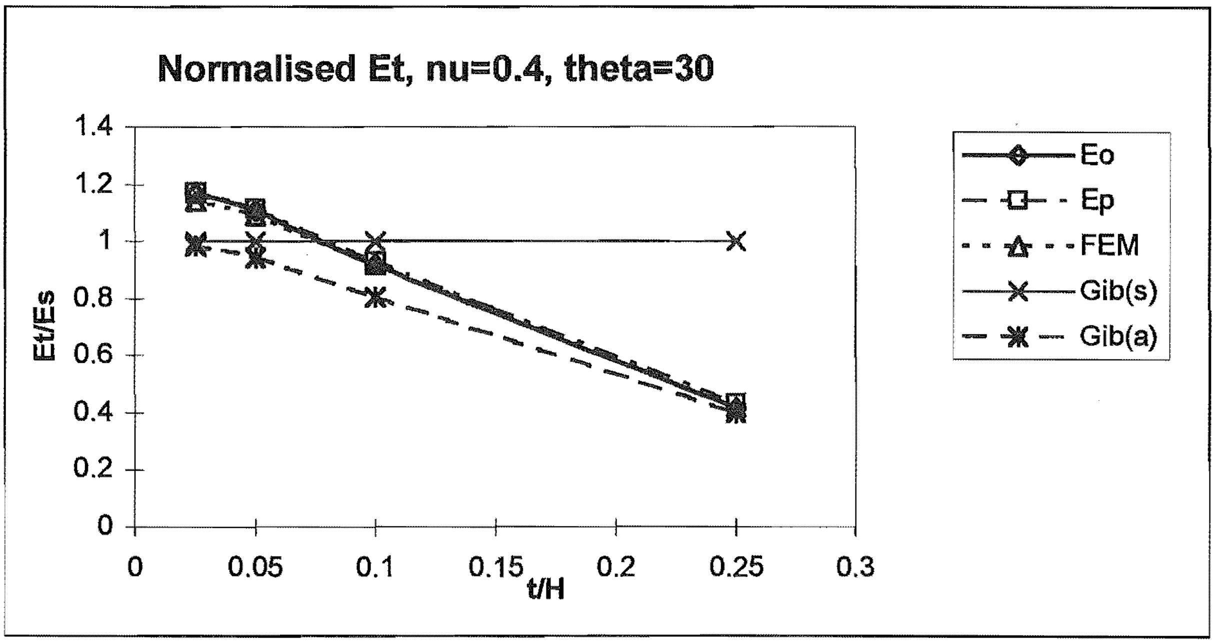
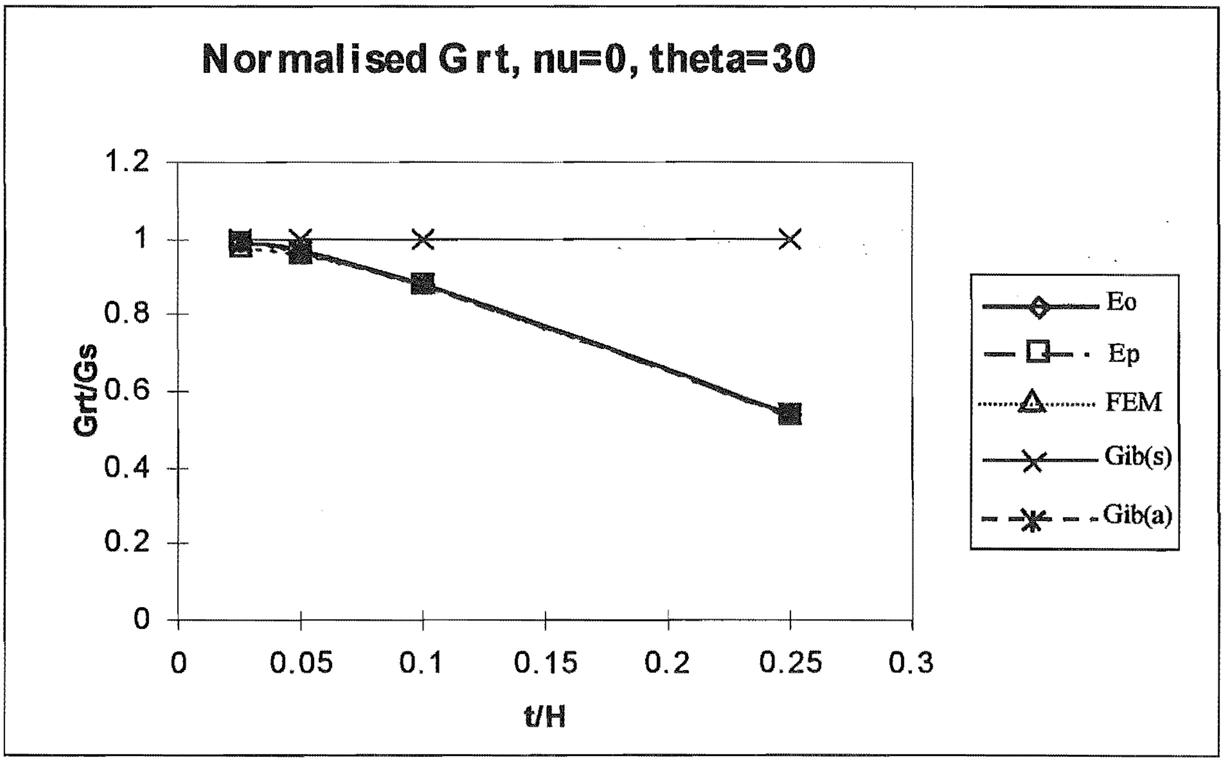


Figure 6.40 Et for cell angle=30, v=0.4



Figur 6.41 Grt for cell angle=30, v=0

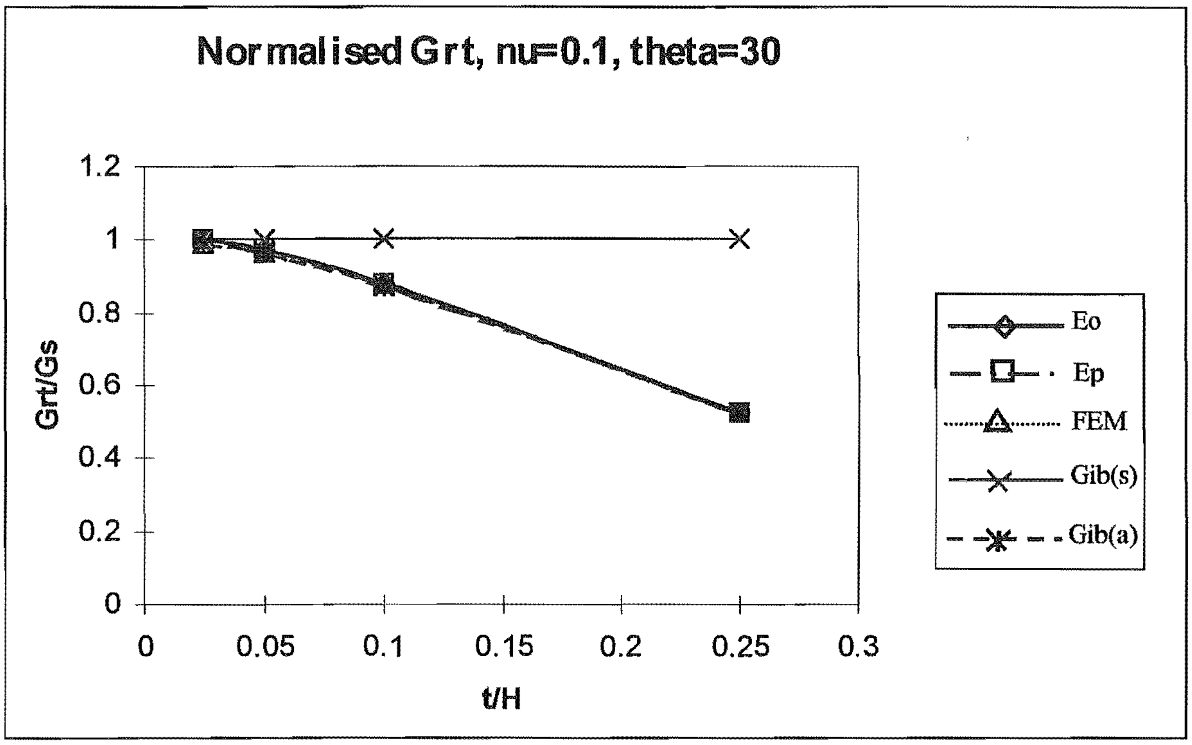


Figure 6.42 Grt for cell angle=30, v=0.1

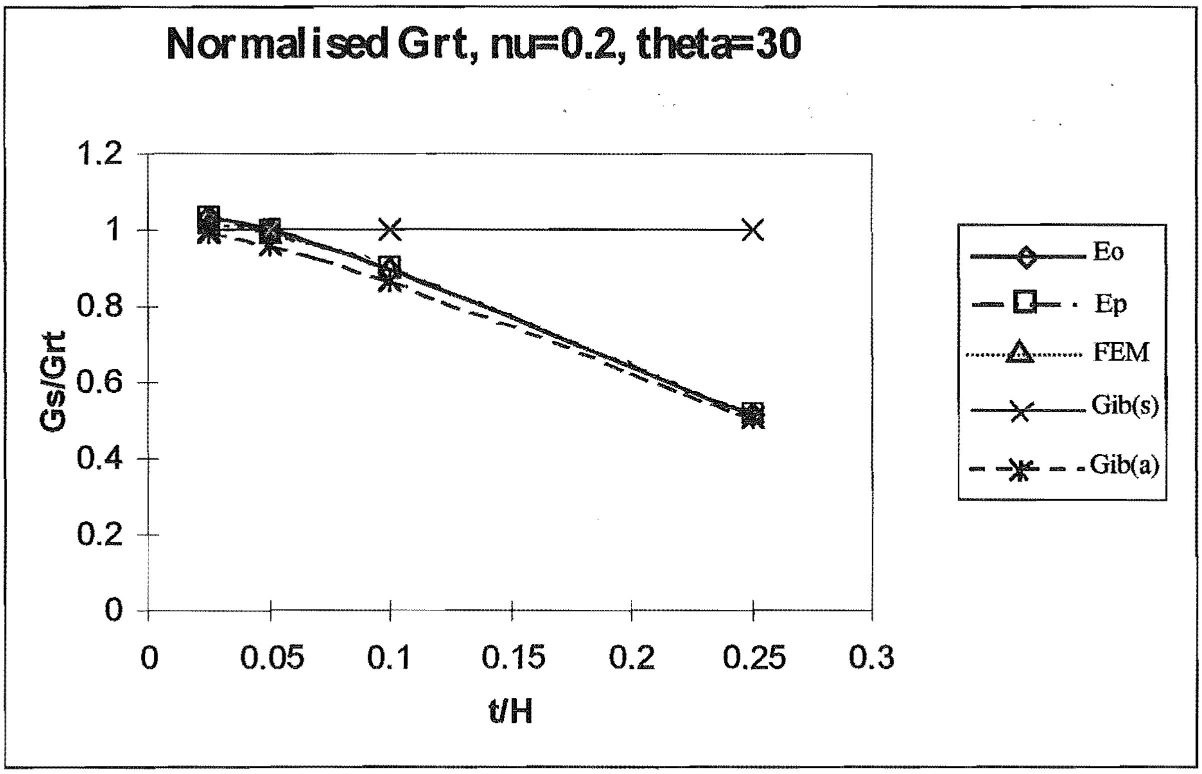


Figure 6.43 Grt for cell angle=30, v=0.2

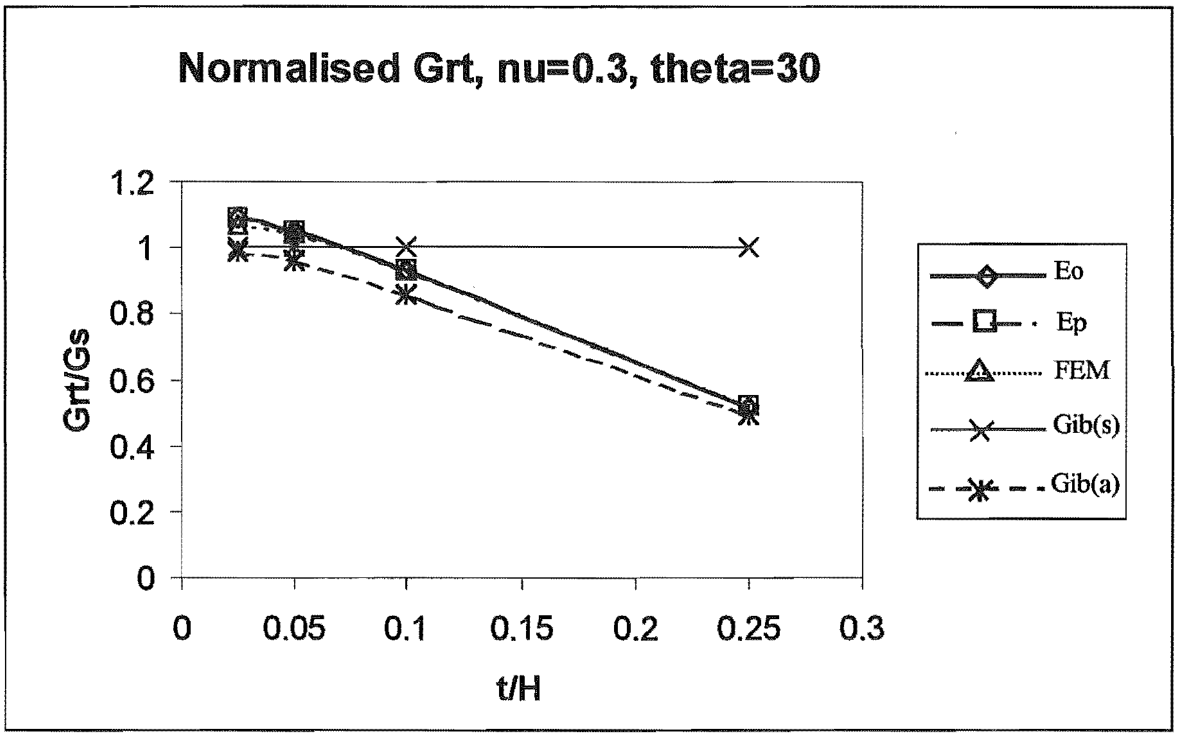


Figure 6.44 Grt for cell angle=30, v=0.3

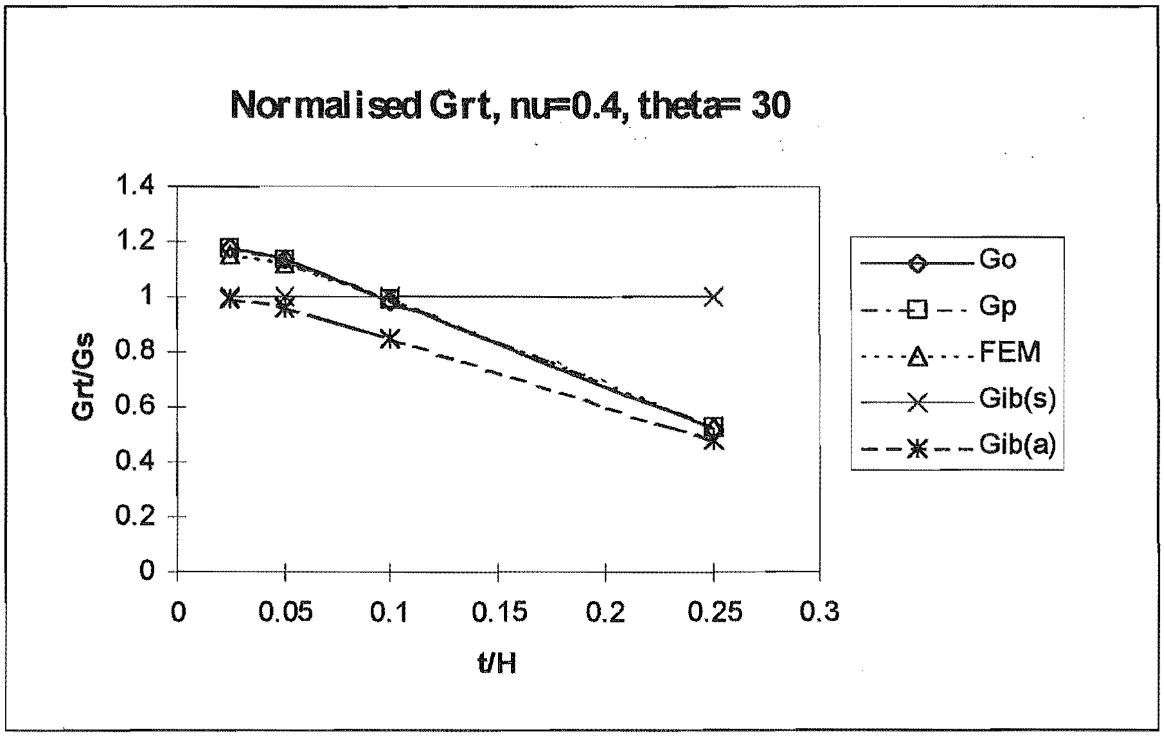


Figure 6.45 Grt for cell angle=30, v=0.4

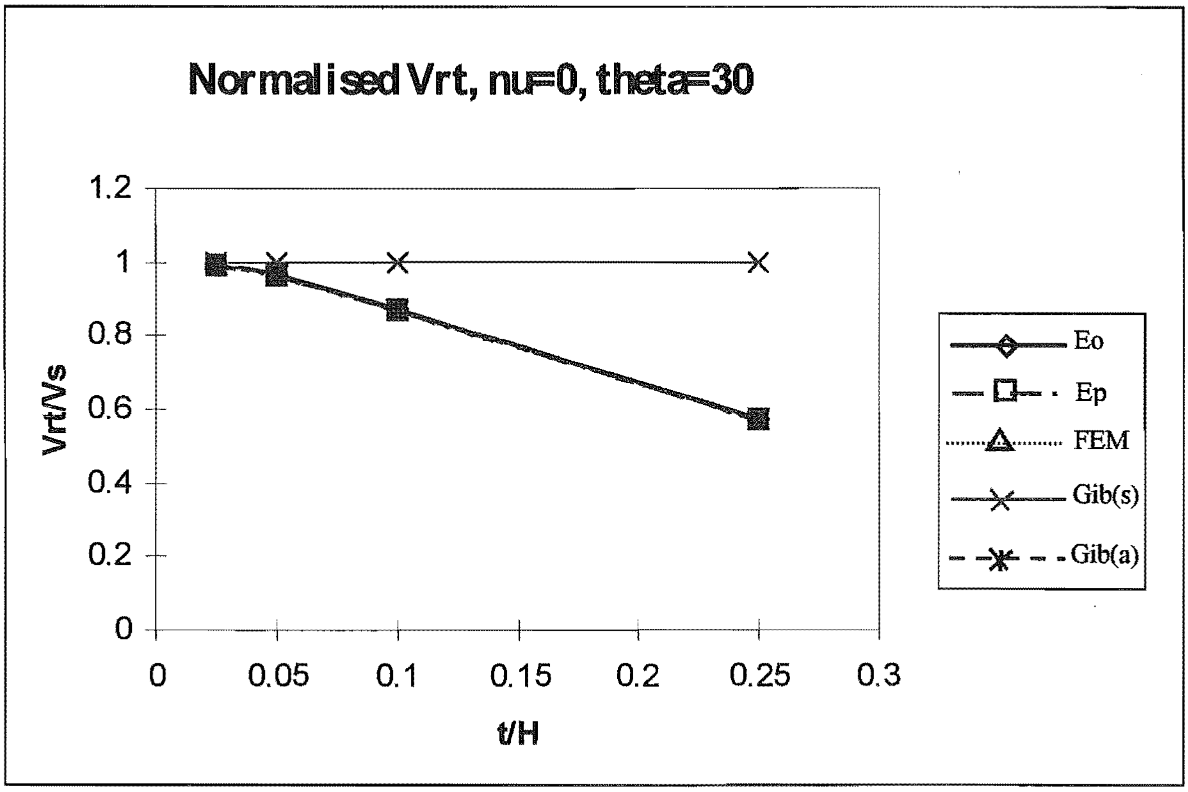


Figure 6.46 Vrt for cell angle=30, v=0

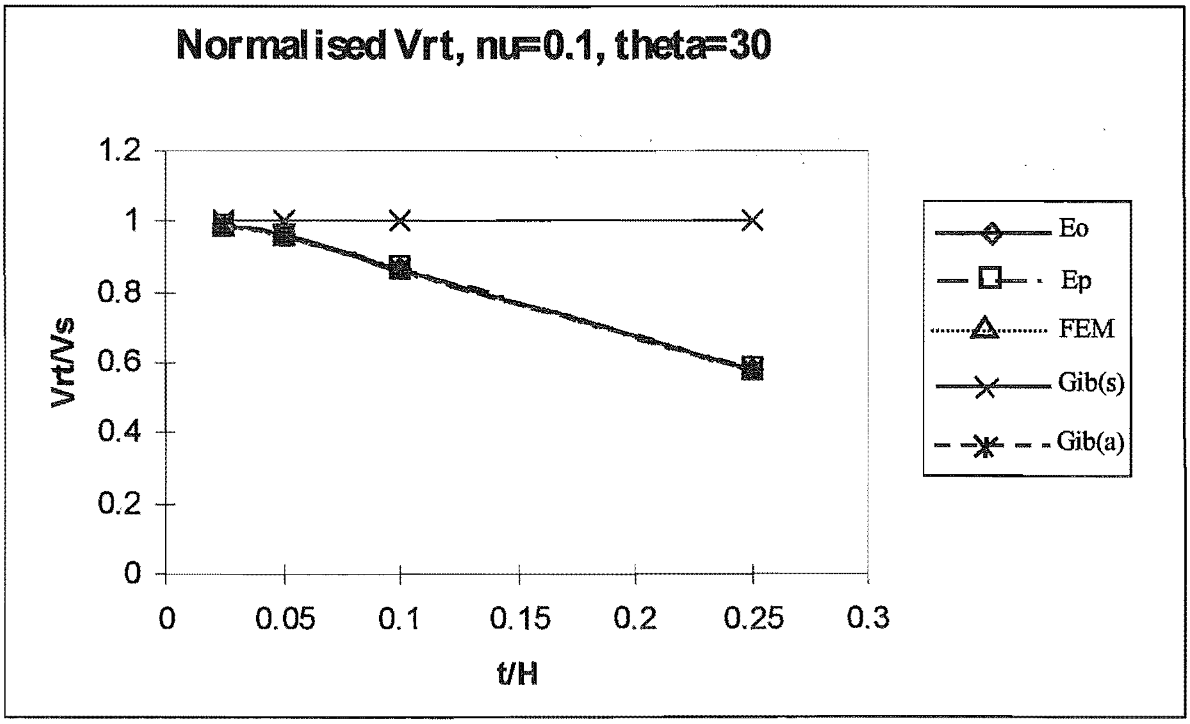


Figure 6.47 Vrt for cell angle=30, v=0.1

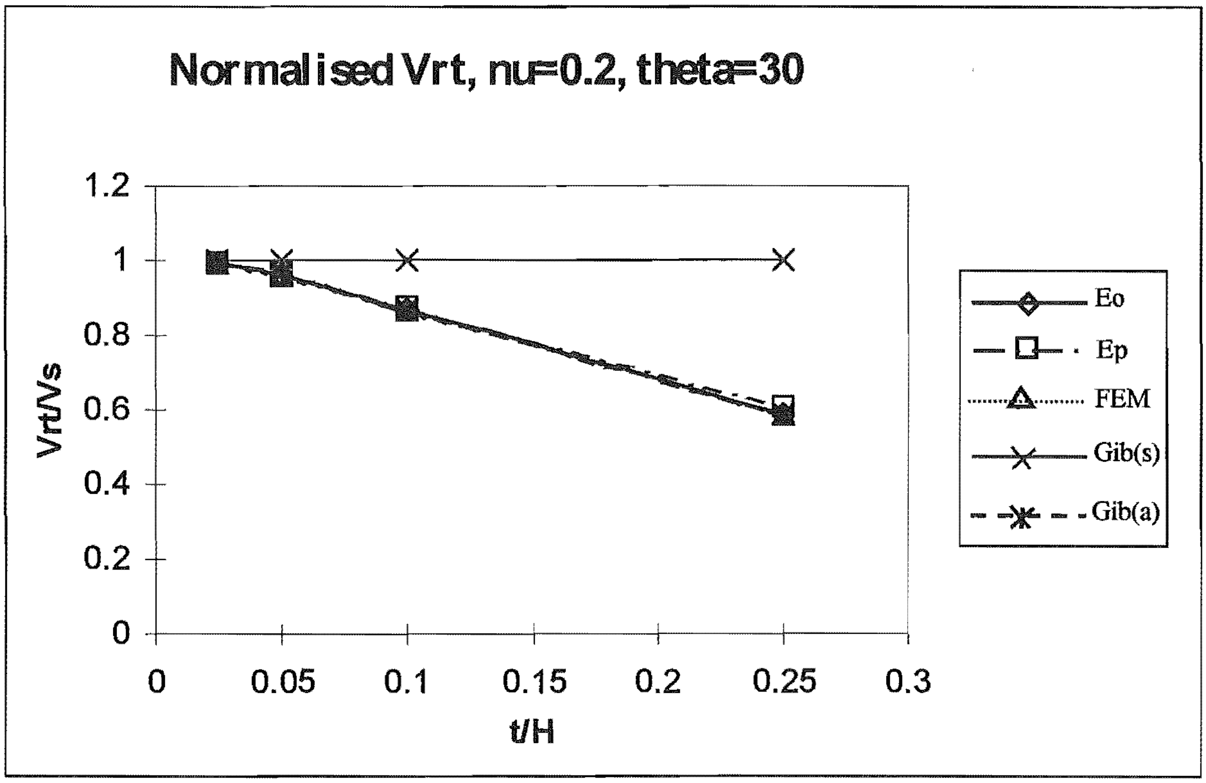


Figure 6.48 Vrt for cell angle=30, v=0.2

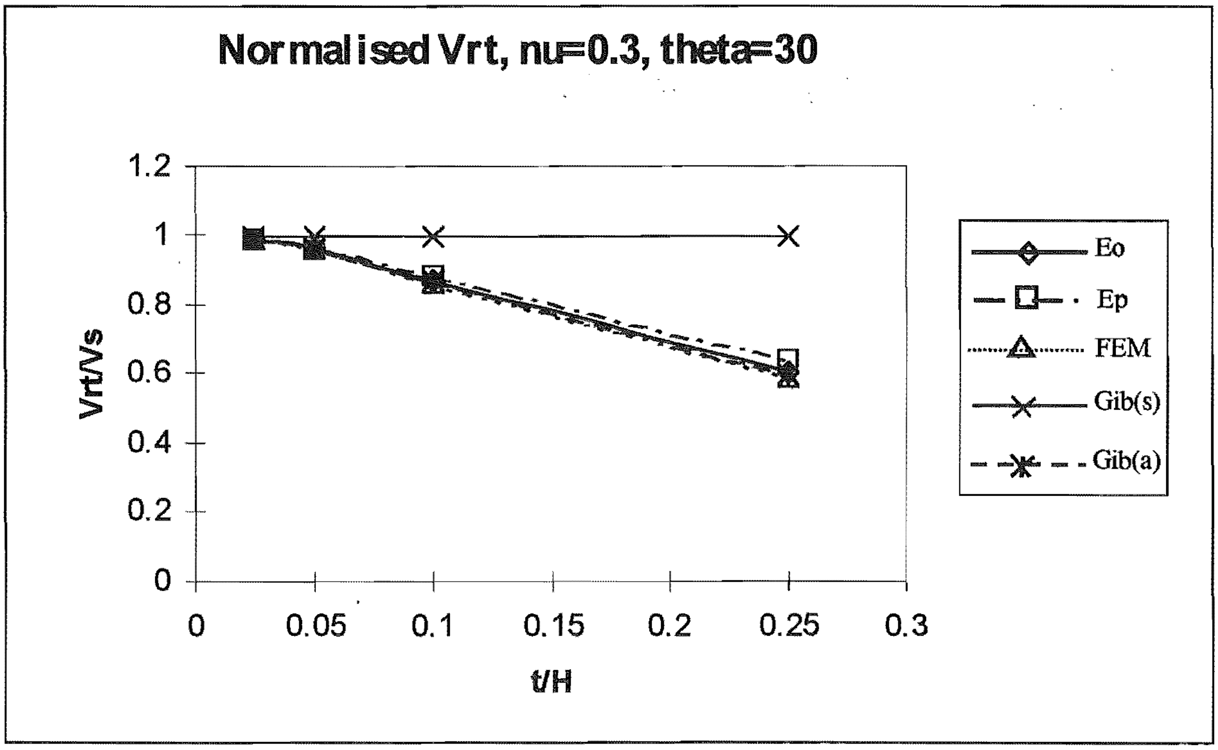


Figure 6.49 Vrt for cell angle=30, v=0.3

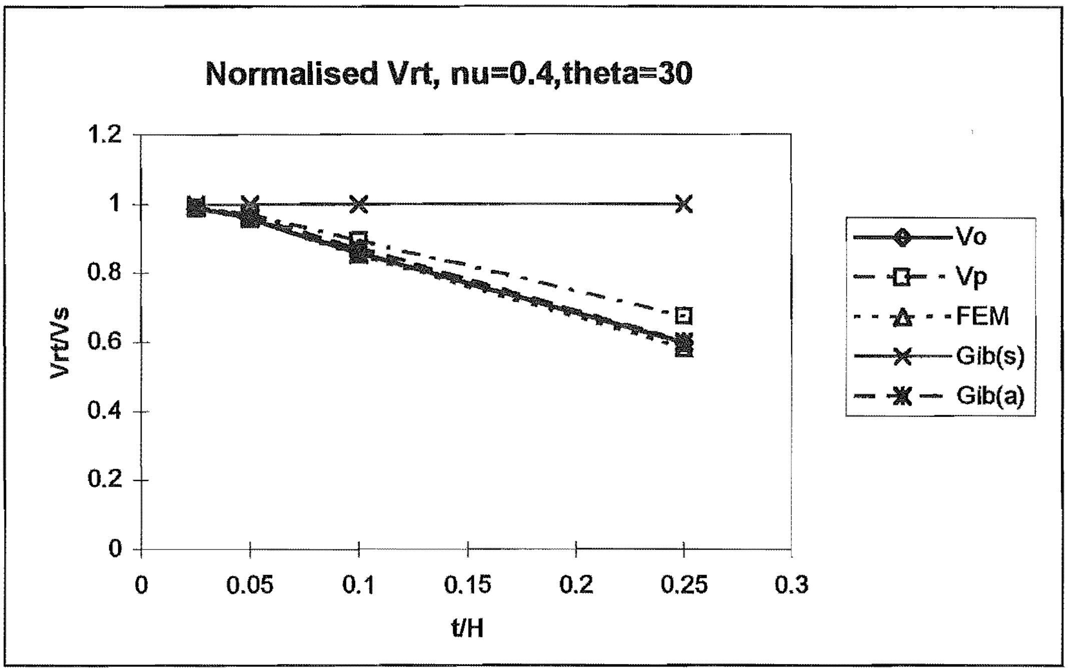


Figure 6.50 Vrt for cell angle=30, v=0.4

Chapter 7

7. Conclusion

The numerical solution extension of Stol's work to the growth ring model is a step towards a boardscale model and is in broad agreement with the experimental result. The challenge is to understand the discrepancies with either a more definitive measurement method or model that can be validated with a high level of certainty or a boardscale model that can accommodate variation of cellular geometries within growth rings. These discrepancies are either experimental measurement errors due to limitation of current existing measurement methology or the lack of consideration of other physical geometrical parameters such as variations in the tangential direction.

The analytical solutions for an isotropic cell wall material using the plate energy method with and without the longitudinal strain effects present a closer approximation than the Gibson's advance equation when compared to the FEM results of regular model taking into consideration the Poisson's effect of plate. The next step is to find an analytical solution that takes into consideration the orthotropic elastic properties and Possion's ratio of the cell wall material into consideration and relate these to the physical microfibril angle of the tracheid.

8. References

[1] Gibson, L.J. and Ashby, M. F.: (1988) Cellular solids; Struture and Properties, Pergamon Oxford.

[2] Stol, K.A. (1995): Finite Element shell Analysis of Cyclical cellular structures with ANSYS 5.0; Final year thesis, University of Canterbury.

[3] Astley, R.J., Stol, K.A. and Harrington, J.J.: (1998) Modelling the elastic properties of softwood. Part I and II, the cellurlar microstructure. Holz als Rohund Werkstoff (in publication)

[4] Winkelmann K, MaCallion H, Evans R and Pugh M,D :(1998). Measuring MFA and elastic properties of pinus radiata wood, in the significance of MFA to wood quality, Canterbury University Press, Christchurch. Wood technology Group.

[5] Harrington, J.J., Booker R, and Astley R.J. :(1998) Modeling the elastic properties of softwood. Part I, the cell wall lamellae. Holz als Rohund Werkstoff (in publication)

[6] Robert Evans: 1994, Rapid measurement of the Transverse dimensions of tracheids in radial wood sections from Pinus Radiata, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia, Holzforschung, volume 48,168-172.

[7] H. Abe and J. Ohtani and K. Fukazawa, 1992: Microfibrillar Orientation of the Innermost Surface of Conifer Tracheid Walls, IAWA Bulletin, Vol 13, Number 4.

[8] J.W.P. Nicholls: 1986. Within-tree variation of wood characteristics of Pinus Radiata. D. Don., CSIRO, Australia.

[9] K. Pernestal, B. Jonsson, B. Larsson 1994: A simple model for density of annual rings. Wood Science and Technology 29 (1995).

[10] K. Stol, 1996: Effect of Tangential wall offset, internal paper. University of Canterbury, Dept of Mechanical Engineering, 1996.

[11] K. Winkelmann, 1997. Lab-report: Four point bending measurements on small wood samples. University of Canterbury, Dept of Mechanical Engineering, 1997.

[12] M. F. Ashby and K. E. Easterling and R. Harrysson,1985: The Fracture and Toughness of Woods, Proc. R. Soc. Lond. A,vol. 398.

[13] R.J. Astley: An introduction to Finite Elements in Solids and Structures. Chapman & Hall, 1992.

[14] Bodig, J. And Jayne: Mechanics of Wood and wood composites. Van Nostrand Reinhold Company, New York 1982.

[15] Kollmann, F. F. P. And Cote, Jr.: W.A. Principles of Wood Science and Technology. I. Solid Wood, Springer-Verlag, Berlin 1968.

[16] Dinwoodie, J, M.: Timber, its nature and behaviour, Van Nostrand Reinhold Company, New York 1981.

[17] Kinnimonth, J. A and Whitehouse 1991: Properties and Uses of New Zealand Radiata Pine, Vol I and II. New Zealand Ministry of Forestry, Rotorua, NZ.

[18] Cave, I.D., 1968: The anisotropic elasticity of the plant cell wall, Wood Science and Technology 2: 268-278.

[19] Cave, I.D., 1969: The longitudinal Young's modulus of Pinus radiata. Wood Science and Technology 3: 40-48.

[20] Cave, I.D. , 1978: Modelling moisture-related mechanical properties of Wood-Part I and II. Wood Science and Technology 12.

[21] Carlsson, H. And Thunell, B.: Shrinkage behaviour of wood drying effect of tensile stresses, Paperi ja Puu. No 7, 1975.

[22] Cave, I. D., 1972: A theory of the shrinkage of wood. Wood Science and technology 6: 284-292.

[23] Cave, I. D., 1972: Swelling of a fibre reinforced composite in which the matrix is water-reactive, Wood Science and Technology 3:40-48.

[24] Chou, P.C., Carleone, J, and Hsu, C.M. 1972: Elastic constants of layer media. Journal of Composite materials 6: 80-93.

[25] J. D. Barrett and A. P. Schniewind, 1973:Three-dimensional Finite Element Models of Cylindrical Wood Fibers, Wood and Fiber, volume 5,number 3,215-225.

[26] Persson K (1997): Modelling of wood properties by micromechanical approach. Report TVSM-3020. Divsion of Structural Mechanics, Lund University.

[27] Koponen, S., Toratti, T and Kanerva, P. (1991): Modelling elastic and shrinkage properties of wood based cell structure, Wood Science and Technology 25: 25-32.

[28] Astley, R.J., Harrington, J.J., Tang, S. And Neumann, J. (1998): "Modelling the influence of microfibril angle on stiffness and shrinkage in radiata pine" in "Microfibril Angle in Wood" (B.G. Butterfield ed), Proceedings IAWA/IUFRO International workshop on the significance of Microfibril angle to wood quality, Westport New Zealand 1997, 272-295.

[29] Woodhouse & Kahle, E., 1994:The influence of cell geometry on the elasticity of Softwood, J. Of Materials Science 29: 1250-1259.

[30] Pernestal K., Jonsson B. , Larsson B., 1994: A simple model for density of annual Rings, Wood Science and Technology 29:441-449.

[31] J.F. Senft and B.A. Bendtsen, 1985: Measuring microfibrillar angles using light microscopy, Wood and Fibre Science, 17 (4) 564- 567.

[32] K. Winkelmann, 1997. Lab-report: Summary of samples necessary to obtain experimental data to validate methodology of modelling elasticity of Pinus Radiata timber over a range of MFA. University of Canterbury, Dept of Mechanical Engineering, 1997.

[33] Walker, J.C.F. and Butterfield, B.G. 1995: The importance of Microfibril angle for the processing industries. NZ Journal of Forestry 40(4) 34-40.

[34] Whitney, J.M., 1987: Structural Analysis of laminated anisotropic plates. Technomic Publishing Co. Inc.

[35] Jones, M. R. 1975: Mechanics of composite materials, chapter 4. Scripta Book Company.

[36] Stol, K.A., 1996: Variation of Wall area ratio across a growth cycle. Internal paper, University of Canterbury, Dept of Mechanical Engineering, MEWG.

[37] Cown, D.J. 1975: Variation of tracheid dimensions in the stem of a 26 years old radiata pine tree, Appita Vol 28 (4) 237-245.

[38] Cown, D.J., Young, G.D. and Kimberley, M.O. 1991: Sprial Grain patterns in plantation grown Pinus Radiata, NZ Journal of Foresty Science 21 (2/3): 206-216.

[39] Donaldson, L.A. 1992: Within and between tree variation in microfibril angle in Pinus radiata, NZ Journal of Forestry Science 22(1):77-86.

[40] Bamber, R.K. 1983: The wood properties of Padiata Pine, Commonwealth Agricultural Bureaux.

[41] Robert Evans, Sue-Anne Stuart and John Van der Touw, 1996: Microfibril Angle Scanning of Increment Cores by X-Ray Diffractometry, Appita.

[42] Robert Evans,Geoffrey Downes, David Menz and Sharee Stringer, 1994:Rapid Measurement of Variation in Tracheid Transverse Dimensions in a Radiata Pine Tree, Appita, vol 48 No 2, 134-138.

[43] Donaldson,L.A., 1991:The Use of Pit Apertures as Windows to Measure Microfibril Angle in Chemical Pulp Fibers,Wood and Fiber Science,vol 23, No 2, 290-295.

[44] Crosby, C.M., Zeeuw, C.D. and Marton, R. 1972:Fibrillar Angle Variation in Red Pine Determined by Senarmont Compensation, Journal Wood Science, Vol 6, 185-195.

[45] Meylan, B.A, 1967: Measurement of microfibril angle by x-ray diffraction, Forest Product Journal, 17 (5):51-58.

[46] Cave, I.D., 1966: Theory of x-ray diffraction method for the measurement of microfibril angle in wood. Forest Product Journal, 16(10):37-42.

[47] Cave, I.D., 1997: Theory of x-ray measurement of microfibril angle in wood. Wood Science Technology. 13(3):143-152.

[48] R.J. Astley, J.J. Harrington and K.A. Stol, 1996: Mechanical Modeling of Wood Microstructure, An engineering approach. Proceeding of IPENZ Annual Conference, 1996, Vol 1, Part 2, 403-408.

[49] Stol K., 1995: Finite Element Shell analysis of Cyclic Cellular Structures with ANSYS 5.0, Internal paper, MEWG, Department of Mechanical Engineering, University of Canterbury.

[50] Swanson Analysis Systems, Inc, 1997: ANSYS User's Manual for Revision 5.2.

[51] Walker, J.C.F and Butterfield, B.G. 1995: The importance fo Microfibril Angle for the processing industries. NZ Journal of Forestry 40(4) 34-40.

[52] Roark, and Young, 1996: Formulas for stress and Strain, McGraw-Hill, London, p 96.

[53] Gibson,L.J., 1981: PHd thesis on Advanced Honeycomb structure, University of Cambridge, 1981.

[54] S. Timoshenko and J. N. Goodier, Theory of Plates and Shells, 1959: elasticity theory, plates and shells, McGraw-Hill.

[55] S. A. Ambartsumyan, 1991: Theory of Anisotropic Plates: Strength, Stability, and Vibrations, Hemisphere Publishing Corporation, New York.

[56] R. F. S. Hearmon, 1961: An Introduction to Applied Anisotropic Elasticity, Oxford University Press.

[57] R. Szilard, 1974: Theory and Analysis of Plates: Classical and Numerical Methods, Prentice-Hall, Inc.

[58] R. C. Tang and N. N. Hsu, 1973: Analysis of the Relationship Between Microstructure and Elastic Properties of the Cell Wall, Journal of Wood and Fiber, Vol. 5,2 pages 139-151.

Appendix:

1. Algorithm of Growth ring model

ANSYS macro algorithm

The algorithm uses SHELL 91 as the FE element. This element supports 9 layer composite cell wall structure. The input module requires growth ring geometry data (ring 14, in this case) to enable the normalized data to reconstruct an idealized model based on the X-ray measurements.

The explanation of the structure of the growth ring macro would enable further refinement of the model to be carried out. Growth ring macros are named with gw(XXX).mac where XXX is a descriptor if a different methodology is used such as X-ray MFA and Lerange integration.

```
! Ansys macro to analyse the 'statistical model' of wood with
! wall offset, alpha =0.65, entire growth ring, modified from
! STOFF.mac, with periodic growth ring- 18 April 97

!/MENU,OFF
/NOPR

!geometry information
*ask,wave,average width (radial diameter),34.28                !width
average
*ask,height,average height,26.53
*ask,mfave,average MFA,32.86                                   !MFA
average
*ask,denave,average density,449.98                             !density
average
*ask,alpha,offset factor-alpha,0.65
*ask,theta,hexgon angle,15.886
*ask,cwidth,width of growth ring in um,1885
```

Construction of the cell starts with initializing all data to zero, cell counter and scale factor to 1.

```
dist=0.                                                        ! start
distance from growth ring
deltax=0.
wold=0.         !width of previous cell
cuma=0          !cumulative area
afactor=1       !scale factor
```

```
i=1           ! cell counter
dist=0        !normalised distance
cw=0  !cumulative cell width
cden=0        !cumulative density
cmu=0 !cumulative mfa
```

The PREP7 begins with the input data for idealized geometrical data, this example is using the Stain-cell MFA information.

```
/PREP7
! cyclic data for stain cell method
 *dim,S2,table,10,11          !array definition for ULIB S2mpnt

!width data (radial)
 *dim,w,table,6,2        !width array
 *set,w(0,0),0,0, 0.75, 0.85,  .884,      1,          !x co-ord
 *set,w(0,1),1,1.12,    .9,   .72,  1.1072,1.12       !y co-ord

 !mfa values, stain cell, AVERAGE= 32.86
  *dim,mfa,table,6,2
  *set,mfa(0,0),0,      0,    0.55, 0.84, 1.    !x co-ord
  *set,mfa(0,1),1,      1.05, 1.05, .83,  1.05


!density

  *dim,d,table,6,2
  *set,d(0,0),0,  0,    0.6,  0.85, 0.884,      1             !x co-ord
  *set,d(0,1),1,  .78,  .9789,      1.5524,     0.8745        ,.78
```

Construction of the first cell uses a scale factor of 1 and the computed geometrical information from above. Cumulative information is calculated as a check on the final model.

```
!data from growth ring 14,CSIRO
/COM, DATA

width=w(dist,1)*wave                      ! width  = cell width
W1=WIDTH
cw=cw+width                     !cell 1 width
cent=width/2                    !centroid position
dist=cent/cwidth                !next cell position
den=d(dist,1)*denave            !density cal
mu=mfa(dist,1)*mfave            !MFA cal
aratior=den/1460                !area ratio cal

cden=cden+den                   !cum density cal
cmu=cmu+mu                      ! cum MFA cal
```

Subroutine gwring1 will define the cell using the scale factor and the computed geometrical data. Cell wall length is computed as well to check on the final area ratio. Subsequent cells are scaled and translated using ARSCALE and ATRAN to the cell position.

```
/COM, SUBROUTINE GWRING1
*USE,gwring1                              ! Custom macro for model
generation

! define origin of entire growth ring
LOCAL,13,0,-width/2*(1+alpha),0,0,0,0    ! Co-ord. system at corner of
model

!calculating line length of cell
*get,lmin,line,,num,min
*GET,LEN1,LINE,1,LENG
*GET,LEN2,LINE,2,LENG
*GET,LEN3,LINE,3,LENG
*GET,LEN4,LINE,4,LENG
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA

csys,0
/com,! generating other elements using ATRAN and ARSCALE

*do,i,2,100

width=w(dist,1)*wave
cw=cw+width

*if,cw,gt,cwidth,then
/com, cell width exceeded
*exit
*endif

! Computing the next cell location
cent=cw-width/2
dist= cent/ cwidth

den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460
cden=cden+den
cmu=cmu+mu

csys,13
*GET,ANUM,AREA,,NUM,MAX                  !GET AREA START NUMBER
*get,xmax,node,,mxloc,x                  !GET START OF NEW CELL CO-ORD,
FROM CSYS13

CLOCAL,14,0,xmax,0,0                     ! Co-ord system for start of new
cell
csys,13

ATRAN,14,1,7,,,1,0                       !ATRAN to start of next cell
SCALE=Width/w1
CSYS,14
*GET,ANUM,AREA,,NUM,MAX                  !GET AREA START NUMBER
ASEL,S,AREA,,ANUM-6,ANUM,,,0             !SELECT NEW AREA FOR ELEMENT
GENERATION
```

```
ARSCALE,ANUM-6,ANUM,,SCALE,,,,1,1
*use,gwring1


!calculating line length of cell
X2N=X2*SCALE
X3N=X3*SCALE
X4N=X4*SCALE
LEN2=SQRT((Y2-Y3)**2+(X2N-X3N)**2)
LEN4=SQRT((Y4-Y2)**2+(X4N-X2N)**2)
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA


!define new cell element type
ASEL,ALL
csys,13

*ENDDO


cw=cw-width
i=i-1        !number of cell
swave=cw/I   !cal ave width
sden=cden/I  !cal ave density
smu=cmu/I    !cal ave MFA

*GET,height,KP,,MXLOC,Y
*GET,width,KP,,MXLOC,X


cuma=cuma/width/height  !cal total area



nummrg,all
numcmp,all
```

Upon completion of the geometrical model, the FE macro applies the cyclic constrains and the loading in various directions, computes the elastic constants and write them to an output file.

```
/com, starting of FEM calculation

NSEL,S,LOC,X,width                          ! Get points where forces act
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!
NSEL,S,LOC,X,width*0.999,width*1.001        ! Get points where forces act
NSEL,U,LOC,X,0                              !  *0.001 avoids default
*0.005
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height*0.999,height*1.001
```

```
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
NSEL,U,LOC,X,0                          ! In case no +ve X-face
exists
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!!!!!!!

/COM, USING CNSTRNTS
*ULIB,CNSTRNTS
/com, passed cnstrnts
/COM, USING NOR_CPLE
*USE,NOR_CPLE                    ! Apply normal load constraints
/COM, PASSED NOR_CPLE
WSORT
FINISH

/SOLU

NSEL,ALL
F,nfixx,FX,1                     ! x-normal loading
SOLVE

FDELE,ALL
F,nfixy,FY,1                     ! y-normal loading
SOLVE

FDELE,ALL
F,nfixz,FZ,1                     ! z-normal loading
SOLVE

FINISH

!!!!!!!!!!!!!!!!!!!! SHEAR STRESS !!!!!!!!!!!!!!!!

/PREP7
LSCLEAR,ALL
CPDELE,ALL                      ! Clear previous loads and
constraints
CEDELE,ALL
*USE,YX_CPLE                    ! Apply yx-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FX,1                    ! yx-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
CPDELE,ALL                      ! Clear previous loads and
constraints
CEDELE,ALL
*USE,YZ_CPLE                    ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FZ,1                    ! yz-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
```

```
CPDELE,ALL                        ! Clear previous loads and
constraints
CEDELE,ALL
*USE,XZ_CPLE                      ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixx,FZ,1                      ! xz-shear loading
ANTYPE,,REST
SOLVE
FINISH


/POST1

SET,1                                     ! Calculates moduli
EX=width/(UX(nfixx)*height*depth)
NUXY=-UY(nfixy)*width/(UX(nfixx)*height)
NUXZ=-UZ(nfixz)*width/(UX(nfixx)*depth)
SET,2
EY=height/(UY(nfixy)*width*depth)
NUYX=-UX(nfixx)*height/(UY(nfixy)*width)
NUYZ=-UZ(nfixz)*height/(UY(nfixy)*depth)
SET,3
EZ=depth/(UZ(nfixz)*height*width)
NUZX=-UX(nfixx)*depth/(UZ(nfixz)*width)
NUZY=-UY(nfixy)*depth/(UZ(nfixz)*height)
SET,4
GYX=height/(UX(nfixy)*width*depth)
SET,5
GYZ=height/(UZ(nfixy)*width*depth)
SET,6
GXZ=width/(UZ(nfixx)*height*depth)


/OUTPUT,gwstain,txt,,APPEND
aave=denave/1460        !cell wall density is assumed as 1460

*vwrite
('')
*vwrite
('***********************gw(stain).mac********************')
*vwrite
('Growth ring 14 analysis, read in data')
*vwrite,wave,mfave,denave
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)
*vwrite,cwidth,aave
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of read in data*****')


*VWRITE
('')
*vwrite
('******calculated data******')
*VWRITE,i
('cell number=',f8.3)
*vwrite,swave,smu,sden
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)

*vwrite,cw,cuma
```

```
('growth ring width=',f8.1,'     area ratio=',f8.3)
*vwrite
('*****end of calculated data*****')

*VWRITE
('ER          ET        EL       VTR              VLR         VLT       .    GTR
             GTL       GRL')
*vwrite,ex,ey,ez,nuyx,nuzx,nuzy,gyx,gyz,gxz     ! Maxm of 10 outputs
(9E11.4)
/OUTPUT,TERM

FINISH
```

The necessary subroutines must be available within the same directory for the macro to access them. The subroutines are:

1. CONSTRNT - cyclic constrain subroutine

2. GWRING1 - element geometrical definition subroutine

3. S2MPT - element wall material properties definition subroutine

Listings of the subroutine macros are found in the appendix of this thesis.

2. ANSYS macro - CNSTRNTS

```
! Library of macros to apply cyclic constraints to any repeated unit,
but
!   assuming no corner nodes exist (o.k. for honeycomb)
! Constraints o.k. for units with no positive x-faces
!
! Constraint macros included for: (Ex, Ey, Ez), Gyx, Gxy, Gyz, Gxz
! Assummed parameters:
!       width = model width
!       height = model height
!       depth = model depth
!       nfixx = node on +ve X-face where force acts
!       nfixy = node on +ve Y-face where force acts
!       nfixz = node on +ve Z-face where force acts
/eof

NOR_CPLE                                 ! CONSTRAINTS FOR NORMAL
LOADING


D,NODE(0,NY(nfixx),NZ(nfixx)),UX         ! Fix nodes
D,NODE(NX(nfixy),0,NZ(nfixy)),UY
D,NODE(NX(nfixz),NY(nfixz),0),UZ

NSEL,S,LOC,Z,depth                       ! Select Z-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Y,0.01,height-0.01
*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE                    ! Meaning: do next node
 NSEL,A,LOC,Z,0
 n2=NODE(NX(n),NY(n),0)
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,ROTX,n,n2
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,NE,nfixz,THEN
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n,UZ,1,n2,UZ,-1,nfixz,UZ,-1      ! Wj'-Wj-Wi=0
 *ENDIF
 NSEL,U,LOC,Z,0
*ENDDO
!C***finished z-face


NSEL,S,LOC,X,width*0.999,width*1.001           ! Select X-face
NSEL,R,LOC,Y,0.01,height-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,A,LOC,X,0
 n2=NODE(0,NY(n),NZ(n))
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UY,n,n2
 CP,cpnum+2,UZ,n,n2
 CP,cpnum+3,ROTX,n,n2
```

```
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,NE,nfixx,THEN
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n,UX,1,n2,UX,-1,nfixx,UX,-1      ! Uj'-Uj-Ui=0
 *ENDIF
NSEL,U,LOC,X,0
*ENDDO
!C***finished x-face


NSEL,S,LOC,Y,height*0.999,height*1.001              ! Select Y-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,A,LOC,Y,0
 n2=NODE(NX(n),0,NZ(n))
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UZ,n,n2
 CP,cpnum+3,ROTX,n,n2
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,NE,nfixy,THEN
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n,UY,1,n2,UY,-1,nfixy,UY,-1      ! Vj'-Vj-Vi=0
 *ENDIF
 NSEL,U,LOC,Y,0
*ENDDO
!C***finished y-face


NSEL,S,LOC,Z,depth                          ! Select Y-edge
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,ALL
 n2=NODE(0,NY(n),0)
 n3=NODE(width,NY(n),0)
 n4=NODE(width,NY(n),depth)
 *IF,NY(n)/1000,EQ,NY(n3)/1000,THEN           ! If +ve x-face exists!
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UY,n,n2,n3,n4
  CP,cpnum+2,ROTX,n,n2,n3,n4
  CP,cpnum+3,ROTY,n,n2,n3,n4
  CP,cpnum+4,ROTZ,n,n2,n3,n4
  *IF,n3,EQ,nfixx,THEN
   D,n,UX
   *GET,cenum,ACTIVE,,CE
   CE,cenum+1,0,n4,UX,1,n3,UX,-1                 ! Have to use CE
  *ELSEIF,n4,EQ,nfixx,THEN
   D,n2,UX
   *GET,cenum,ACTIVE,,CE
   CE,cenum+1,0,n4,UX,1,n3,UX,-1
  *ELSE
   *GET,cenum,ACTIVE,,CE
```

```
    CE,cenum+1,0,n2,UX,1,n,UX,-1                   ! Couple
    CE,cenum+2,0,n4,UX,1,n,UX,-1,nfixx,UX,-1       ! Constraint
    CE,cenum+3,0,n3,UX,1,n,UX,-1,nfixx,UX,-1       ! Constraint
   *ENDIF
   *GET,cenum,ACTIVE,,CE
   CE,cenum+1,0,n3,UZ,1,n2,UZ,-1                  ! Couple
   CE,cenum+2,0,n,UZ,1,n2,UZ,-1,nfixz,UZ,-1       ! Constraint
   CE,cenum+3,0,n4,UZ,1,n2,UZ,-1,nfixz,UZ,-1      ! Constraint
  *ELSE                                           ! If no +ve x-face
exists
   *GET,cpnum,ACTIVE,,CP
   CP,cpnum+1,UX,n,n2
   CP,cpnum+2,UY,n,n2
   CP,cpnum+3,ROTX,n,n2
   CP,cpnum+4,ROTY,n,n2
   CP,cpnum+5,ROTZ,n,n2
   *GET,cenum,ACTIVE,,CE
   CE,cenum+1,0,n,UZ,1,n2,UZ,-1,nfixz,UZ,-1       ! Constraint
  *ENDIF
  NSEL,S,LOC,Z,depth
  NSEL,R,LOC,X,0
  NSEL,R,LOC,Y,0.01,height-0.01
*ENDDO
!C***finished y-edge


NSEL,S,LOC,Z,depth                  ! Select X-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,ALL
 n2=NODE(NX(n),0,0)
 n3=NODE(NX(n),height,0)
 n4=NODE(NX(n),height,depth)
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *IF,n3,EQ,nfixy,THEN
  D,n,UY
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n4,UY,1,n3,UY,-1                    ! Couple
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n2,UY,1,n,UY,-1                   ! Couple
  CE,cenum+2,0,n4,UY,1,n,UY,-1,nfixy,UY,-1       ! Constraint
  CE,cenum+3,0,n3,UY,1,n,UY,-1,nfixy,UY,-1       ! Constraint
 *ENDIF
 *IF,n,EQ,nfixz,THEN
  D,n3,UZ
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n4,UZ,1,n,UZ,-1                   ! Couple
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n3,UZ,1,n2,UZ,-1                  ! Couple
  CE,cenum+2,0,n,UZ,1,n2,UZ,-1,nfixz,UZ,-1       ! Constraint
  CE,cenum+3,0,n4,UZ,1,n2,UZ,-1,nfixz,UZ,-1      ! Constraint
 *ENDIF
 NSEL,S,LOC,Z,depth
 NSEL,R,LOC,Y,0
```

```
 NSEL,R,LOC,X,0.01,width-0.01
*ENDDO
!C***finished x-edge


NSEL,S,LOC,X,0                              ! Select Z-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,ALL
 n2=NODE(width,0,NZ(n))
 n3=NODE(width,height,NZ(n))
 n4=NODE(0,height,NZ(n))
 *IF,NY(n)/1000,EQ,NY(n2)/1000,THEN            ! If +ve x-face exists!
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UZ,n,n2,n3,n4
  CP,cpnum+2,ROTX,n,n2,n3,n4
  CP,cpnum+3,ROTY,n,n2,n3,n4
  CP,cpnum+4,ROTZ,n,n2,n3,n4
  *GET,cenum,ACTIVE,,CE
  *IF,n3,EQ,nfixy,THEN
   D,n,UY
   CE,cenum+1,0,n4,UY,1,n3,UY,-1                  ! Couple
  *ELSEIF,n4,EQ,nfixy,THEN
   D,n2,UY
   CE,cenum+1,0,n4,UY,1,n3,UY,-1                  ! Couple
  *ELSE
   CE,cenum+1,0,n2,UY,1,n,UY,-1                  ! Couple
   CE,cenum+2,0,n4,UY,1,n,UY,-1,nfixy,UY,-1      ! Constraint
   CE,cenum+3,0,n3,UY,1,n,UY,-1,nfixy,UY,-1      ! Constraint
  *ENDIF
  *GET,cenum,ACTIVE,,CE
  *IF,n2,EQ,nfixx,THEN
   D,n4,UX
   CE,cenum+1,0,n2,UX,1,n3,UX,-1                  ! Couple
  *ELSEIF,n3,EQ,nfixx,THEN
   D,n,UX
   CE,cenum+1,0,n2,UX,1,n3,UX,-1                  ! Couple
  *ELSE
   CE,cenum+1,0,n4,UX,1,n,UX,-1                   ! Couple
   CE,cenum+2,0,n3,UX,1,n,UX,-1,nfixx,UX,-1       ! Constraint
   CE,cenum+3,0,n2,UX,1,n,UX,-1,nfixx,UX,-1       ! Constraint
  *ENDIF
 *ELSE                                          ! If no +ve x-face
exists
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UX,n,n4
  CP,cpnum+2,UZ,n,n4
  CP,cpnum+3,ROTX,n,n4
  CP,cpnum+4,ROTY,n,n4
  CP,cpnum+5,ROTZ,n,n4
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n4,UY,1,n,UY,-1,nfixy,UY,-1      ! Constraint
 *ENDIF
 NSEL,S,LOC,X,0
 NSEL,R,LOC,Y,0
 NSEL,R,LOC,Z,0.01,depth-0.01
*ENDDO
!C***finished z-edge

NSEL,S,LOC,X,0                              ! Select corners
```

```
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0

*GET,numsel,NODE,,COUNT
NSEL,ALL

*IF,numsel,NE,0,THEN
 n=NODE(0,0,0)
 n2=NODE(0,0,depth)
 n3=NODE(width,0,depth)
 n4=NODE(width,0,0)
 n5=NODE(0,height,0)
 n6=NODE(0,height,depth)
 n7=NODE(width,height,depth)
 n8=NODE(width,height,0)
 *IF,NY(n)/1000,EQ,NY(n4)/1000,THEN           ! If +ve x-face exists!
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,ROTX,n,n2,n3,n4,n5,n6,n7,n8
  CP,cpnum+2,ROTY,n,n2,n3,n4,n5,n6,n7,n8
  CP,cpnum+3,ROTZ,n,n2,n3,n4,n5,n6,n7,n8
  *GET,cenum,ACTIVE,,CE
  *IF,n3,EQ,nfixx,THEN
   D,n,UX
   D,n5,UX
   D,n6,UX
   CE,cenum+1,0,n4,UX,1,n3,UX,-1               ! Couple UX
   CE,cenum+2,0,n7,UX,1,n3,UX,-1               ! Couple UX
   CE,cenum+3,0,n8,UX,1,n3,UX,-1               ! Couple UX
  *ELSEIF,n4,EQ,nfixx,THEN
   D,n2,UX
   D,n5,UX
   D,n6,UX
   CE,cenum+1,0,n3,UX,1,n4,UX,-1               ! Couple UX
   CE,cenum+2,0,n7,UX,1,n4,UX,-1               ! Couple UX
   CE,cenum+3,0,n8,UX,1,n4,UX,-1               ! Couple UX
  *ELSEIF,n7,EQ,nfixx,THEN
   D,n,UX
   D,n2,UX
   D,n5,UX
   CE,cenum+1,0,n3,UX,1,n7,UX,-1               ! Couple UX
   CE,cenum+2,0,n4,UX,1,n7,UX,-1               ! Couple UX
   CE,cenum+3,0,n8,UX,1,n7,UX,-1               ! Couple UX
  *ELSEIF,n8,EQ,nfixx,THEN
   D,n,UX
   D,n2,UX
   D,n6,UX
   CE,cenum+1,0,n3,UX,1,n8,UX,-1               ! Couple UX
   CE,cenum+2,0,n4,UX,1,n8,UX,-1               ! Couple UX
   CE,cenum+3,0,n7,UX,1,n8,UX,-1               ! Couple UX
  *ELSE
   *GET,cenum,ACTIVE,,CE
   CE,cenum+1,0,n4,UX,1,n,UX,-1,nfixx,UX,-1     ! Constraint
   CE,cenum+2,0,n3,UX,1,n2,UX,-1,nfixx,UX,-1    ! Constraint
   CE,cenum+3,0,n7,UX,1,n6,UX,-1,nfixx,UX,-1    ! Constraint
   CE,cenum+4,0,n8,UX,1,n5,UX,-1,nfixx,UX,-1    ! Constraint
  *ENDIF
  *GET,cenum,ACTIVE,,CE
  *IF,n5,EQ,nfixy,THEN
   D,n2,UY
   D,n3,UY
   D,n4,UY
   CE,cenum+1,0,n6,UY,1,n5,UY,-1               ! Couple UY
   CE,cenum+2,0,n7,UY,1,n5,UY,-1               ! Couple UY
   CE,cenum+3,0,n8,UY,1,n5,UY,-1               ! Couple UY
  *ELSEIF,n6,EQ,nfixy,THEN
   D,n,UY
```

```
  D,n3,UY
  D,n4,UY
  CE,cenum+1,0,n5,UY,1,n6,UY,-1                    ! Couple UY
  CE,cenum+2,0,n7,UY,1,n6,UY,-1                    ! Couple UY
  CE,cenum+3,0,n8,UY,1,n6,UY,-1                    ! Couple UY
 *ELSEIF,n7,EQ,nfixy,THEN
  D,n,UY
  D,n2,UY
  D,n4,UY
  CE,cenum+1,0,n5,UY,1,n7,UY,-1                    ! Couple UY
  CE,cenum+2,0,n6,UY,1,n7,UY,-1                    ! Couple UY
  CE,cenum+3,0,n8,UY,1,n7,UY,-1                    ! Couple UY
 *ELSEIF,n8,EQ,nfixy,THEN
  D,n,UY
  D,n2,UY
  D,n3,UY
  CE,cenum+1,0,n5,UY,1,n8,UY,-1                    ! Couple UY
  CE,cenum+2,0,n6,UY,1,n8,UY,-1                    ! Couple UY
  CE,cenum+3,0,n7,UY,1,n8,UY,-1                    ! Couple UY
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n5,UY,1,n,UY,-1,nfixy,UY,-1        ! Constraint
  CE,cenum+2,0,n6,UY,1,n2,UY,-1,nfixy,UY,-1        ! Constraint
  CE,cenum+3,0,n7,UY,1,n3,UY,-1,nfixy,UY,-1        ! Constraint
  CE,cenum+4,0,n8,UY,1,n4,UY,-1,nfixy,UY,-1        ! Constraint
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n2,EQ,nfixz,THEN
  D,n4,UZ
  D,n5,UZ
  D,n8,UZ
  CE,cenum+1,0,n3,UZ,1,n2,UZ,-1                    ! Couple UZ
  CE,cenum+2,0,n6,UZ,1,n2,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n2,UZ,-1                    ! Couple UZ
 *ELSEIF,n3,EQ,nfixz,THEN
  D,n,UZ
  D,n5,UZ
  D,n8,UZ
  CE,cenum+1,0,n2,UZ,1,n3,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n6,UZ,1,n3,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n3,UZ,-1                    ! Couple UZ
 *ELSEIF,n6,EQ,nfixz,THEN
  D,n,UZ
  D,n4,UZ
  D,n8,UZ
  CE,cenum+1,0,n2,UZ,1,n6,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n3,UZ,1,n6,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n6,UZ,-1                    ! Couple UZ
 *ELSEIF,n7,EQ,nfixz,THEN
  D,n,UZ
  D,n4,UZ
  D,n5,UZ
  CE,cenum+1,0,n2,UZ,1,n7,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n3,UZ,1,n7,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n6,UZ,1,n7,UZ,-1                    ! Couple UZ
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n2,UZ,1,n,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+2,0,n3,UZ,1,n4,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+3,0,n6,UZ,1,n5,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+4,0,n7,UZ,1,n8,UZ,-1,nfixz,UZ,-1        ! Constraint
 *ENDIF
*ELSE                                             ! If no +ve x-face
exists
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2,n5,n6
```

```
  CP,cpnum+2,ROTX,n,n2,n5,n6
  CP,cpnum+3,ROTY,n,n2,n5,n6
  CP,cpnum+4,ROTZ,n,n2,n5,n6
  *GET,cenum,ACTIVE,,CE
  !!!!!!!!!! CE,cenum+1,0,n4,UY,1,n,UY,-1,nfixy,UY,-1      !
Constraint
 *ENDIF
*ENDIF
!C***finished corners

/EOF


YX_CPLE                          ! CONSTRAINTS FOR YX SHEAR

! width = model width
! height = model height
! depth = model depth
! Co-ord. system at corner of model


D,NODE(0,NY(nfixx),NZ(nfixx)),UY        ! Fix nodes
D,NODE(NX(nfixy),0,NZ(nfixy)),UX
D,NODE(NX(nfixz),NY(nfixz),0),UZ

NSEL,S,LOC,Z,depth                      ! Select Z-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,A,LOC,Z,0
 n2=NODE(NX(n),NY(n),0)
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,ROTX,n,n2
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,EQ,nfixz,THEN
  D,n,UZ
 *ELSE
  CP,cpnum+6,UZ,n,n2
 *ENDIF
 NSEL,U,LOC,Z,0
*ENDDO
!C***finished z-face


NSEL,S,LOC,X,width*0.999,width*1.001          ! Select X-face
NSEL,R,LOC,Y,0.01,height-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,A,LOC,X,0
 n2=NODE(0,NY(n),NZ(n))
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UZ,n,n2
```

```
 CP,cpnum+3,ROTX,n,n2
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,EQ,nfixx,THEN
  D,n,UY
 *ELSE
  CP,cpnum+6,UY,n,n2
 *ENDIF
 NSEL,U,LOC,X,0
*ENDDO
!C***finished x-face

NSEL,S,LOC,Y,height*0.999,height*1.001                ! Select Y-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
 NSEL,A,LOC,Y,0
 n2=NODE(NX(n),0,NZ(n))
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UY,n,n2
 CP,cpnum+2,UZ,n,n2
 CP,cpnum+3,ROTX,n,n2
 CP,cpnum+4,ROTY,n,n2
 CP,cpnum+5,ROTZ,n,n2
 *IF,n,NE,nfixy,THEN
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n,UX,1,n2,UX,-1,nfixy,UX,-1
 *ENDIF
 NSEL,U,LOC,Y,0
*ENDDO
!C***finished y-face

NSEL,S,LOC,Z,depth                  ! Select Y-edge
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(0,NY(n),0)
n3=NODE(width,NY(n),0)
n4=NODE(width,NY(n),depth)
*IF,NY(n)/1000,EQ,NY(n3)/1000,THEN           ! If +ve x-face exists
 *GET,cpnum,ACTIVE,,CP                       !   tolerance of 1.0E-7
 CP,cpnum+1,UX,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *IF,n3,EQ,nfixx,THEN
  D,n,UY
  D,n3,UY
  D,n4,UY
 *ELSEIF,n4,EQ,nfixx,THEN
  D,n2,UY
  D,n3,UY
  D,n4,UY
```

```
  *ELSE
   CP,cpnum+5,UY,n,n2,n3,n4
  *ENDIF
  *IF,n,EQ,nfixz,THEN
   D,n,UZ
   D,n3,UZ
   D,n4,UZ
  *ELSEIF,n4,EQ,nfixz,THEN
   D,n,UZ
   D,n2,UZ
   D,n4,UZ
  *ELSE
   CP,cpnum+6,UZ,n,n2,n3,n4
  *ENDIF
*ELSE
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,UZ,n,n2                ! Hopefully n will never equal nfixz
 CP,cpnum+4,ROTX,n,n2
 CP,cpnum+5,ROTY,n,n2
 CP,cpnum+6,ROTZ,n,n2
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01
*ENDDO
!C***finished y-edge


NSEL,S,LOC,Z,depth                   ! Select X-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(NX(n),0,0)
n3=NODE(NX(n),height,0)
n4=NODE(NX(n),height,depth)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2,n3,n4
CP,cpnum+2,ROTX,n,n2,n3,n4
CP,cpnum+3,ROTY,n,n2,n3,n4
CP,cpnum+4,ROTZ,n,n2,n3,n4
*IF,n,EQ,nfixz,THEN
D,n,UZ
D,n3,UZ
D,n4,UZ
*ELSEIF,n4,EQ,nfixz,THEN
D,n,UZ
D,n2,UZ
D,n4,UZ
*ELSE
CP,cpnum+5,UZ,n,n2,n3,n4
*ENDIF
*GET,cenum,ACTIVE,,CE
*IF,n3,EQ,nfixy,THEN
D,n,UX
CE,cenum+1,0,n4,UX,1,n3,UX,-1                ! Couple
*ELSEIF,n4,EQ,nfixy,THEN
D,n2,UX
CE,cenum+1,0,n4,UX,1,n3,UX,-1                ! Couple
```

```
*ELSE
CE,cenum+1,0,n2,UX,1,n,UX,-1                    ! Couple
CE,cenum+2,0,n3,UX,1,n,UX,-1,nfixy,UX,-1        ! Constraint
CE,cenum+3,0,n4,UX,1,n,UX,-1,nfixy,UX,-1        ! Constraint
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01
*ENDDO
!C***finished x-edge


NSEL,S,LOC,X,0                            ! Select Z-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(width,0,NZ(n))
n3=NODE(width,height,NZ(n))
n4=NODE(0,height,NZ(n))
*IF,NY(n)/1000,EQ,NY(n2)/1000,THEN              ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UZ,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *GET,cenum,ACTIVE,,CE
 *IF,n3,EQ,nfixy,THEN
  D,n,UX
  CE,cenum+1,0,n4,UX,1,n3,UX,-1                   ! Couple
 *ELSEIF,n4,EQ,nfixy,THEN
  D,n2,UX
  CE,cenum+1,0,n4,UX,1,n3,UX,-1                   ! Couple
 *ELSE
  CE,cenum+1,0,n2,UX,1,n,UX,-1                  ! Couple
  CE,cenum+2,0,n4,UX,1,n,UX,-1,nfixy,UX,-1      ! Constraint
  CE,cenum+3,0,n3,UX,1,n,UX,-1,nfixy,UX,-1      ! Constraint
 *ENDIF
 *IF,n2,EQ,nfixx,THEN
  D,n2,UY
  D,n3,UY
  D,n4,UY
 *ELSEIF,n3,EQ,nfixx,THEN
  D,n,UY
  D,n2,UY
  D,n3,UY
 *ELSE
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UY,n,n2,n3,n4
 *ENDIF
*ELSE                                           ! If no +ve x-face
exists
!!!!!!!!!!!!!!!!!!
*ENDIF
NSEL,S,LOC,X,0
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01
*ENDDO
!C***finished z-edge

NSEL,S,LOC,X,0                            ! Select corners
```

```
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0

*GET,numsel,NODE,,COUNT
NSEL,ALL

*IF,numsel,NE,0,THEN
n=NODE(0,0,0)
n2=NODE(0,0,depth)
n3=NODE(width,0,depth)
n4=NODE(width,0,0)
n5=NODE(0,height,0)
n6=NODE(0,height,depth)
n7=NODE(width,height,depth)
n8=NODE(width,height,0)
NSEL,S,NODE,,n
NSEL,A,NODE,,n2
NSEL,A,NODE,,n3
NSEL,A,NODE,,n4
NSEL,A,NODE,,n5
NSEL,A,NODE,,n6
NSEL,A,NODE,,n7
NSEL,A,NODE,,n8
*IF,NY(n)/1000,EQ,NY(n4)/1000,THEN                 ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,ROTX,ALL
 CP,cpnum+2,ROTY,ALL
 CP,cpnum+3,ROTZ,ALL
 *IF,n3,EQ,nfixx,THEN
  D,ALL,UY
 *ELSEIF,n4,EQ,nfixx,THEN
  D,ALL,UY
 *ELSEIF,n7,EQ,nfixx,THEN
  D,ALL,UY
 *ELSEIF,n8,EQ,nfixx,THEN
  D,ALL,UY
 *ELSE
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UY,ALL
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n5,EQ,nfixy,THEN
  D,n2,UX
  D,n3,UX
  D,n4,UX
  CE,cenum+1,0,n6,UX,1,n5,UX,-1                       ! Couple UX
  CE,cenum+2,0,n7,UX,1,n5,UX,-1                       ! Couple UX
  CE,cenum+3,0,n8,UX,1,n5,UX,-1                       ! Couple UX
 *ELSEIF,n6,EQ,nfixy,THEN
  D,n,UX
  D,n3,UX
  D,n4,UX
  CE,cenum+1,0,n5,UX,1,n6,UX,-1                       ! Couple UX
  CE,cenum+2,0,n7,UX,1,n6,UX,-1                       ! Couple UX
  CE,cenum+3,0,n8,UX,1,n6,UX,-1                       ! Couple UX
 *ELSEIF,n7,EQ,nfixy,THEN
  D,n,UX
  D,n2,UX
  D,n4,UX
  CE,cenum+1,0,n5,UX,1,n7,UX,-1                       ! Couple UX
  CE,cenum+2,0,n6,UX,1,n7,UX,-1                       ! Couple UX
  CE,cenum+3,0,n8,UX,1,n7,UX,-1                       ! Couple UX
 *ELSEIF,n8,EQ,nfixy,THEN
  D,n,UX
  D,n2,UX
  D,n3,UX
```

```
  CE,cenum+1,0,n5,UX,1,n8,UX,-1                    ! Couple UX
  CE,cenum+2,0,n6,UX,1,n8,UX,-1                    ! Couple UX
  CE,cenum+3,0,n7,UX,1,n8,UX,-1                    ! Couple UX
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n5,UX,1,n,UX,-1,nfixy,UX,-1        ! Constraint
  CE,cenum+2,0,n6,UX,1,n2,UX,-1,nfixy,UX,-1        ! Constraint
  CE,cenum+3,0,n7,UX,1,n3,UX,-1,nfixy,UX,-1        ! Constraint
  CE,cenum+4,0,n8,UX,1,n4,UX,-1,nfixy,UX,-1        ! Constraint
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n2,EQ,nfixz,THEN
  D,n4,UZ
  D,n5,UZ
  D,n8,UZ
  CE,cenum+1,0,n3,UZ,1,n2,UZ,-1                    ! Couple UZ
  CE,cenum+2,0,n6,UZ,1,n2,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n2,UZ,-1                    ! Couple UZ
 *ELSEIF,n3,EQ,nfixz,THEN
  D,n,UZ
  D,n5,UZ
  D,n8,UZ
  CE,cenum+1,0,n2,UZ,1,n3,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n6,UZ,1,n3,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n3,UZ,-1                    ! Couple UZ
 *ELSEIF,n6,EQ,nfixz,THEN
  D,n,UZ
  D,n4,UZ
  D,n8,UZ
  CE,cenum+1,0,n2,UZ,1,n6,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n3,UZ,1,n6,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n6,UZ,-1                    ! Couple UZ
 *ELSEIF,n7,EQ,nfixz,THEN
  D,n,UZ
  D,n4,UZ
  D,n5,UZ
  CE,cenum+1,0,n2,UZ,1,n7,UZ,-1                     ! Couple UZ
  CE,cenum+2,0,n3,UZ,1,n7,UZ,-1                    ! Couple UZ
  CE,cenum+3,0,n6,UZ,1,n7,UZ,-1                    ! Couple UZ
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n2,UZ,1,n,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+2,0,n3,UZ,1,n4,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+3,0,n6,UZ,1,n5,UZ,-1,nfixz,UZ,-1        ! Constraint
  CE,cenum+4,0,n7,UZ,1,n8,UZ,-1,nfixz,UZ,-1        ! Constraint
 *ENDIF
*ELSE                                                          -face
exists
!!!!!!!!!!
*ENDIF
*ENDIF
NSEL,ALL
!C***finished corners

/EOF


XY_CPLE                              !  CONSTRAINTS FOR XY SHEAR
                                     !   Yet to be tested !!!!!!!!!!!!!

D,NODE(0,NY(nfixx),NZ(nfixx)),UY          ! Fix nodes
D,NODE(NX(nfixy),0,NZ(nfixy)),UX
D,NODE(NX(nfixz),NY(nfixz),0),UZ

NSEL,S,LOC,Z,depth                   ! Select Z-face
NSEL,R,LOC,X,0.01,width-0.01
```

```
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Z,0
n2=NODE(NX(n),NY(n),0)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UY,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixz,THEN
D,n,UZ
*ELSE
CP,cpnum+6,UZ,n,n2
*ENDIF
NSEL,U,LOC,Z,0
*ENDDO
!C***finished z-face


NSEL,S,LOC,X,width*0.999,width*1.001                ! Select X-face
NSEL,R,LOC,Y,0.01,height-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,X,0
n2=NODE(0,NY(n),NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,NE,nfixx,THEN
*GET,cenum,ACTIVE,,CE
CE,cenum+1,0,n,UY,1,n2,UY,-1,nfixx,UY,-1
*ENDIF
NSEL,U,LOC,X,0
*ENDDO
!C***finished x-face


NSEL,S,LOC,Y,height*0.999,height*1.001              ! Select Y-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Y,0
n2=NODE(NX(n),0,NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
```

```
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixy,THEN
D,n,UX
*ELSE
CP,cpnum+6,UX,n,n2
*ENDIF
NSEL,U,LOC,Y,0
*ENDDO
!C***finished y-face


NSEL,S,LOC,Z,depth                    ! Select Y-edge
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(0,NY(n),0)
n3=NODE(width,NY(n),0)
n4=NODE(width,NY(n),depth)
*IF,NY(n)/1000,EQ,NY(n3)/1000,THEN          ! If +ve x-face exists
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *IF,n,EQ,nfixz,THEN
  D,n,UZ
  D,n3,UZ
  D,n4,UZ
 *ELSEIF,n4,EQ,nfixz,THEN
  D,n,UZ
  D,n2,UZ
  D,n4,UZ
 *ELSE
  CP,cpnum+5,UZ,n,n2,n3,n4
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n3,EQ,nfixx,THEN
  D,n,UY
  CE,cenum+1,0,n4,UY,1,n3,UY,-1                ! Couple!
 *ELSEIF,n4,EQ,nfixx,THEN
  D,n2,UY
  CE,cenum+1,0,n4,UY,1,n3,UY,-1                ! Couple
 *ELSE
  CE,cenum+1,0,n2,UY,1,n,UY,-1                ! Couple
  CE,cenum+2,0,n3,UY,1,n,UY,-1,nfixx,UY,-1    ! Constraint
  CE,cenum+3,0,n4,UY,1,n,UY,-1,nfixx,UY,-1    ! Constraint
 *ENDIF
*ELSE
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,UZ,n,n2
 CP,cpnum+4,ROTX,n,n2
 CP,cpnum+5,ROTY,n,n2
 CP,cpnum+6,ROTZ,n,n2
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01
```

```
*ENDDO
!C***finished y-edge

NSEL,S,LOC,Z,depth                        ! Select X-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(NX(n),0,0)
n3=NODE(NX(n),height,0)
n4=NODE(NX(n),height,depth)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2,n3,n4
CP,cpnum+2,ROTX,n,n2,n3,n4
CP,cpnum+3,ROTY,n,n2,n3,n4
CP,cpnum+4,ROTZ,n,n2,n3,n4
*IF,n3,EQ,nfixy,THEN
D,n,UX
D,n3,UX
D,n4,UX
*ELSEIF,n4,EQ,nfixy,THEN
D,n2,UX
D,n3,UX
D,n4,UX
*ELSE
CP,cpnum+5,UX,n,n2,n3,n4
*ENDIF
*IF,n,EQ,nfixz,THEN
D,n,UZ
D,n3,UZ
D,n4,UZ
*ELSEIF,n4,EQ,nfixz,THEN
D,n,UZ
D,n2,UZ
D,n4,UZ
*ELSE
CP,cpnum+6,UZ,n,n2,n3,n4
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01
*ENDDO
!C***finished x-edge

/EOF


YZ_CPLE                                      ! CONSTRAINTS FOR YZ SHEAR


D,NODE(0,NY(nfixx),NZ(nfixx)),UY             ! Fix nodes
D,NODE(NX(nfixy),0,NZ(nfixy)),UZ
D,NODE(NX(nfixz),NY(nfixz),0),UX

NSEL,S,LOC,Z,depth                           ! Select Z-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN
```

```
*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Z,0
n2=NODE(NX(n),NY(n),0)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixz,THEN
 D,n,UX
*ELSE
 CP,cpnum+6,UX,n,n2
*ENDIF
NSEL,U,LOC,Z,0
*ENDDO
!C***finished z-face

NSEL,S,LOC,X,width*0.999,width*1.001                  ! Select X-face
NSEL,R,LOC,Y,0.01,height-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,X,0
n2=NODE(0,NY(n),NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixx,THEN
 D,n,UY
*ELSE
 CP,cpnum+6,UY,n,n2
*ENDIF
NSEL,U,LOC,X,0
*ENDDO
!C***finished x-face

NSEL,S,LOC,Y,height*0.999,height*1.001                ! Select Y-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Y,0
n2=NODE(NX(n),0,NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UY,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,NE,nfixy,THEN
```

```
 *GET,cenum,ACTIVE,,CE
 CE,cenum+1,0,n,UZ,1,n2,UZ,-1,nfixy,UZ,-1
*ENDIF
NSEL,U,LOC,Y,0
*ENDDO
!C***finished y-face


NSEL,S,LOC,Z,depth                  ! Select Y-edge
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(0,NY(n),0)
n3=NODE(width,NY(n),0)
n4=NODE(width,NY(n),depth)
*IF,NY(n)/1000,EQ,NY(n3)/1000,THEN         ! If +ve x-face exists
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UZ,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *IF,n3,EQ,nfixx,THEN
  D,n,UY
  D,n3,UY
  D,n4,UY
 *ELSEIF,n4,EQ,nfixx,THEN
  D,n2,UY
  D,n3,UY
  D,n4,UY
 *ELSE
  CP,cpnum+5,UY,n,n2,n3,n4
 *ENDIF
 *IF,n,EQ,nfixz,THEN
  D,n,UX
  D,n3,UX
  D,n4,UX
 *ELSEIF,n4,EQ,nfixz,THEN
  D,n,UX
  D,n2,UX
  D,n4,UX
 *ELSE
  CP,cpnum+6,UX,n,n2,n3,n4
 *ENDIF
*ELSE
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,UZ,n,n2
 CP,cpnum+4,ROTX,n,n2
 CP,cpnum+5,ROTY,n,n2
 CP,cpnum+6,ROTZ,n,n2
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01
*ENDDO
!C***finished y-edge


NSEL,S,LOC,Z,depth                  ! Select X-edge
```

```
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(NX(n),0,0)
n3=NODE(NX(n),height,0)
n4=NODE(NX(n),height,depth)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2,n3,n4
CP,cpnum+2,ROTX,n,n2,n3,n4
CP,cpnum+3,ROTY,n,n2,n3,n4
CP,cpnum+4,ROTZ,n,n2,n3,n4
*IF,n,EQ,nfixz,THEN
D,n,UX
D,n3,UX
D,n4,UX
*ELSEIF,n4,EQ,nfixz,THEN
D,n,UX
D,n2,UX
D,n4,UX
*ELSE
CP,cpnum+5,UX,n,n2,n3,n4
*ENDIF
*GET,cenum,ACTIVE,,CE
*IF,n3,EQ,nfixy,THEN
D,n,UZ
CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                     ! Couple
*ELSEIF,n4,EQ,nfixy,THEN
D,n2,UZ
CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                     ! Couple
*ELSE
CE,cenum+1,0,n2,UZ,1,n,UZ,-1                     ! Couple
CE,cenum+2,0,n3,UZ,1,n,UZ,-1,nfixy,UZ,-1         ! Constraint
CE,cenum+3,0,n4,UZ,1,n,UZ,-1,nfixy,UZ,-1         ! Constraint
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01
*ENDDO
!C***finished x-edge


NSEL,S,LOC,X,0                               ! Select Z-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(width,0,NZ(n))
n3=NODE(width,height,NZ(n))
n4=NODE(0,height,NZ(n))
*IF,NY(n)/1000,EQ,NY(n2)/1000,THEN             ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
```

```
 *GET,cenum,ACTIVE,,CE
 *IF,n3,EQ,nfixy,THEN
  D,n,UZ
  CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                ! Couple
 *ELSEIF,n4,EQ,nfixy,THEN
  D,n2,UZ
  CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                ! Couple
 *ELSE
  CE,cenum+1,0,n2,UZ,1,n,UZ,-1                ! Couple
  CE,cenum+2,0,n4,UZ,1,n,UZ,-1,nfixy,UZ,-1    ! Constraint
  CE,cenum+3,0,n3,UZ,1,n,UZ,-1,nfixy,UZ,-1    ! Constraint
 *ENDIF
 *IF,n2,EQ,nfixx,THEN
  D,n2,UY
  D,n3,UY
  D,n4,UY
 *ELSEIF,n3,EQ,nfixx,THEN
  D,n,UY
  D,n2,UY
  D,n3,UY
 *ELSE
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UY,n,n2,n3,n4
 *ENDIF
*ELSE                                         ! If no +ve x-face
exists
!!!!!!!!!!!!!!!!!!
*ENDIF
NSEL,S,LOC,X,0
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01
*ENDDO
!C***finished z-edge

NSEL,S,LOC,X,0                          ! Select corners
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0

*GET,numsel,NODE,,COUNT
NSEL,ALL

*IF,numsel,NE,0,THEN
n=NODE(0,0,0)
n2=NODE(0,0,depth)
n3=NODE(width,0,depth)
n4=NODE(width,0,0)
n5=NODE(0,height,0)
n6=NODE(0,height,depth)
n7=NODE(width,height,depth)
n8=NODE(width,height,0)
NSEL,S,NODE,,n
NSEL,A,NODE,,n2
NSEL,A,NODE,,n3
NSEL,A,NODE,,n4
NSEL,A,NODE,,n5
NSEL,A,NODE,,n6
NSEL,A,NODE,,n7
NSEL,A,NODE,,n8
*IF,NY(n)/1000,EQ,NY(n4)/1000,THEN           ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,ROTX,ALL
 CP,cpnum+2,ROTY,ALL
 CP,cpnum+3,ROTZ,ALL
 *IF,n3,EQ,nfixx,THEN
  D,ALL,UY
 *ELSEIF,n4,EQ,nfixx,THEN
```

```
 D,ALL,UY
*ELSEIF,n7,EQ,nfixx,THEN
 D,ALL,UY
*ELSEIF,n8,EQ,nfixx,THEN
 D,ALL,UY
*ELSE
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UY,ALL
*ENDIF
*GET,cenum,ACTIVE,,CE
*IF,n5,EQ,nfixy,THEN
 D,n2,UZ
 D,n3,UZ
 D,n4,UZ
 CE,cenum+1,0,n6,UZ,1,n5,UZ,-1                    ! Couple UZ
 CE,cenum+2,0,n7,UZ,1,n5,UZ,-1                    ! Couple UZ
 CE,cenum+3,0,n8,UZ,1,n5,UZ,-1                    ! Couple UZ
*ELSEIF,n6,EQ,nfixy,THEN
 D,n,UZ
 D,n3,UZ
 D,n4,UZ
 CE,cenum+1,0,n5,UZ,1,n6,UZ,-1                    ! Couple UZ
 CE,cenum+2,0,n7,UZ,1,n6,UZ,-1                    ! Couple UZ
 CE,cenum+3,0,n8,UZ,1,n6,UZ,-1                    ! Couple UZ
*ELSEIF,n7,EQ,nfixy,THEN
 D,n,UZ
 D,n2,UZ
 D,n4,UZ
 CE,cenum+1,0,n5,UZ,1,n7,UZ,-1                    ! Couple UZ
 CE,cenum+2,0,n6,UZ,1,n7,UZ,-1                    ! Couple UZ
 CE,cenum+3,0,n8,UZ,1,n7,UZ,-1                    ! Couple UZ
*ELSEIF,n8,EQ,nfixy,THEN
 D,n,UZ
 D,n2,UZ
 D,n3,UZ
 CE,cenum+1,0,n5,UZ,1,n8,UZ,-1                    ! Couple UZ
 CE,cenum+2,0,n6,UZ,1,n8,UZ,-1                    ! Couple UZ
 CE,cenum+3,0,n7,UZ,1,n8,UZ,-1                    ! Couple UZ
*ELSE
 *GET,cenum,ACTIVE,,CE
 CE,cenum+1,0,n5,UZ,1,n,UZ,-1,nfixy,UZ,-1        ! Constraint
 CE,cenum+2,0,n6,UZ,1,n2,UZ,-1,nfixy,UZ,-1        ! Constraint
 CE,cenum+3,0,n7,UZ,1,n3,UZ,-1,nfixy,UZ,-1        ! Constraint
 CE,cenum+4,0,n8,UZ,1,n4,UZ,-1,nfixy,UZ,-1        ! Constraint
*ENDIF
*GET,cenum,ACTIVE,,CE
*IF,n2,EQ,nfixz,THEN
 D,n4,UX
 D,n5,UX
 D,n8,UX
 CE,cenum+1,0,n3,UX,1,n2,UX,-1                    ! Couple UX
 CE,cenum+2,0,n6,UX,1,n2,UX,-1                    ! Couple UX
 CE,cenum+3,0,n7,UX,1,n2,UX,-1                    ! Couple UX
*ELSEIF,n3,EQ,nfixz,THEN
 D,n,UX
 D,n5,UX
 D,n8,UX
 CE,cenum+1,0,n2,UX,1,n3,UX,-1                    ! Couple UX
 CE,cenum+2,0,n6,UX,1,n3,UX,-1                    ! Couple UX
 CE,cenum+3,0,n7,UX,1,n3,UX,-1                    ! Couple UX
*ELSEIF,n6,EQ,nfixz,THEN
 D,n,UX
 D,n4,UX
 D,n8,UX
 CE,cenum+1,0,n2,UX,1,n6,UX,-1                    ! Couple UX
 CE,cenum+2,0,n3,UX,1,n6,UX,-1                    ! Couple UX
```

```
  CE,cenum+3,0,n7,UX,1,n6,UX,-1                    ! Couple UX
 *ELSEIF,n7,EQ,nfixz,THEN
  D,n,UX
  D,n4,UX
  D,n5,UX
  CE,cenum+1,0,n2,UX,1,n7,UX,-1                    ! Couple UX
  CE,cenum+2,0,n3,UX,1,n7,UX,-1                    ! Couple UX
  CE,cenum+3,0,n6,UX,1,n7,UX,-1                    ! Couple UX
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n2,UX,1,n,UX,-1,nfixz,UX,-1        ! Constraint
  CE,cenum+2,0,n3,UX,1,n4,UX,-1,nfixz,UX,-1       ! Constraint
  CE,cenum+3,0,n6,UX,1,n5,UX,-1,nfixz,UX,-1       ! Constraint
  CE,cenum+4,0,n7,UX,1,n8,UX,-1,nfixz,UX,-1       ! Constraint
 *ENDIF
*ELSE                                              ! If no +ve x-face
exists
!!!!!!!!!!
*ENDIF
*ENDIF
NSEL,ALL
!C***finished corners

/EOF


XZ_CPLE                                    ! CONSTRAINTS FOR XZ SHEAR


D,NODE(0,NY(nfixx),NZ(nfixx)),UZ           ! Fix nodes
D,NODE(NX(nfixy),0,NZ(nfixy)),UX
D,NODE(NX(nfixz),NY(nfixz),0),UY

NSEL,S,LOC,Z,depth                         ! Select Z-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Z,0
n2=NODE(NX(n),NY(n),0)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixz,THEN
D,n,UY
*ELSE
CP,cpnum+6,UY,n,n2
*ENDIF
NSEL,U,LOC,Z,0
*ENDDO
!C***finished z-face


NSEL,S,LOC,X,width*0.999,width*1.001             ! Select X-face
NSEL,R,LOC,Y,0.01,height-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN
```

```
*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,X,0
n2=NODE(0,NY(n),NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UX,n,n2
CP,cpnum+2,UY,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,NE,nfixx,THEN
*GET,cenum,ACTIVE,,CE
CE,cenum+1,0,n,UZ,1,n2,UZ,-1,nfixx,UZ,-1
*ENDIF
NSEL,U,LOC,X,0
*ENDDO
!C***finished x-face


NSEL,S,LOC,Y,height*0.999,height*1.001             ! Select Y-face
NSEL,R,LOC,X,0.01,width-0.01
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,A,LOC,Y,0
n2=NODE(NX(n),0,NZ(n))
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UY,n,n2
CP,cpnum+2,UZ,n,n2
CP,cpnum+3,ROTX,n,n2
CP,cpnum+4,ROTY,n,n2
CP,cpnum+5,ROTZ,n,n2
*IF,n,EQ,nfixy,THEN
D,n,UX
*ELSE
CP,cpnum+6,UX,n,n2
*ENDIF
NSEL,U,LOC,Y,0
*ENDDO
!C***finished y-face


NSEL,S,LOC,Z,depth                  ! Select Y-edge
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(0,NY(n),0)
n3=NODE(width,NY(n),0)
n4=NODE(width,NY(n),depth)
*IF,NY(n)/1000,EQ,NY(n3)/1000,THEN           ! If +ve x-face exists
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
```

```
  *IF,n,EQ,nfixz,THEN
   D,n,UY
   D,n3,UY
   D,n4,UY
  *ELSEIF,n4,EQ,nfixz,THEN
   D,n,UY
   D,n2,UY
   D,n4,UY
  *ELSE
   CP,cpnum+5,UY,n,n2,n3,n4
  *ENDIF
  *GET,cenum,ACTIVE,,CE
  *IF,n3,EQ,nfixx,THEN
   D,n,UZ
   CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                    ! Couple
  *ELSEIF,n4,EQ,nfixx,THEN
   D,n2,UZ
   CE,cenum+1,0,n4,UZ,1,n3,UZ,-1                    ! Couple
  *ELSE
   CE,cenum+1,0,n2,UZ,1,n,UZ,-1                    ! Couple
   CE,cenum+2,0,n3,UZ,1,n,UZ,-1,nfixx,UZ,-1        ! Constraint
   CE,cenum+3,0,n4,UZ,1,n,UZ,-1,nfixx,UZ,-1        ! Constraint
  *ENDIF
*ELSE
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UX,n,n2
 CP,cpnum+2,UY,n,n2
 CP,cpnum+3,UZ,n,n2
 CP,cpnum+4,ROTX,n,n2
 CP,cpnum+5,ROTY,n,n2
 CP,cpnum+6,ROTZ,n,n2
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,X,0
NSEL,R,LOC,Y,0.01,height-0.01
*ENDDO
!C***finished y-edge


NSEL,S,LOC,Z,depth                            ! Select X-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(NX(n),0,0)
n3=NODE(NX(n),height,0)
n4=NODE(NX(n),height,depth)
*GET,cpnum,ACTIVE,,CP
CP,cpnum+1,UZ,n,n2,n3,n4
CP,cpnum+2,ROTX,n,n2,n3,n4
CP,cpnum+3,ROTY,n,n2,n3,n4
CP,cpnum+4,ROTZ,n,n2,n3,n4
*IF,n3,EQ,nfixy,THEN
D,n,UX
D,n3,UX
D,n4,UX
*ELSEIF,n4,EQ,nfixy,THEN
D,n2,UX
D,n3,UX
D,n4,UX
*ELSE
```

```
CP,cpnum+5,UX,n,n2,n3,n4
*ENDIF
*IF,n,EQ,nfixz,THEN
D,n,UY
D,n3,UY
D,n4,UY
*ELSEIF,n4,EQ,nfixz,THEN
D,n,UY
D,n2,UY
D,n4,UY
*ELSE
CP,cpnum+6,UY,n,n2,n3,n4
*ENDIF
NSEL,S,LOC,Z,depth
NSEL,R,LOC,Y,0
NSEL,R,LOC,X,0.01,width-0.01
*ENDDO
!C***finished x-edge


NSEL,S,LOC,X,0                              ! Select Z-edge
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01

*GET,nmax,NODE,,NUM,MAX
*GET,nmin,NODE,,NUM,MIN

*DO,n,nmin,nmax
*IF,NSEL(n),NE,1,CYCLE
NSEL,ALL
n2=NODE(width,0,NZ(n))
n3=NODE(width,height,NZ(n))
n4=NODE(0,height,NZ(n))
*IF,NY(n)/1000,EQ,NY(n2)/1000,THEN            ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,UY,n,n2,n3,n4
 CP,cpnum+2,ROTX,n,n2,n3,n4
 CP,cpnum+3,ROTY,n,n2,n3,n4
 CP,cpnum+4,ROTZ,n,n2,n3,n4
 *GET,cenum,ACTIVE,,CE
 *IF,n3,EQ,nfixy,THEN
  D,n2,UX
  D,n3,UX
  D,n4,UX
 *ELSEIF,n4,EQ,nfixy,THEN
  D,n2,UX
  D,n3,UX
  D,n4,UX
 *ELSE
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UX,n,n2,n3,n4
 *ENDIF
 *IF,n2,EQ,nfixx,THEN
  D,n4,UZ
  CE,cenum+1,0,n2,UZ,1,n3,UZ,-1                 ! Couple
 *ELSEIF,n3,EQ,nfixx,THEN
  D,n,UZ
  CE,cenum+1,0,n2,UZ,1,n3,UZ,-1                 ! Couple
 *ELSE
  CE,cenum+1,0,n4,UZ,1,n,UZ,-1                 ! Couple
  CE,cenum+2,0,n2,UZ,1,n,UZ,-1,nfixx,UZ,-1     ! Constraint
  CE,cenum+3,0,n3,UZ,1,n,UZ,-1,nfixx,UZ,-1     ! Constraint
 *ENDIF
*ELSE                                          ! If no +ve x-face
exists
!!!!!!!!!!!!!!!!!!
```

```
*ENDIF
NSEL,S,LOC,X,0
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0.01,depth-0.01
*ENDDO
!C***finished z-edge

NSEL,S,LOC,X,0                           ! Select corners
NSEL,R,LOC,Y,0
NSEL,R,LOC,Z,0

*GET,numsel,NODE,,COUNT
NSEL,ALL

*IF,numsel,NE,0,THEN
n=NODE(0,0,0)
n2=NODE(0,0,depth)
n3=NODE(width,0,depth)
n4=NODE(width,0,0)
n5=NODE(0,height,0)
n6=NODE(0,height,depth)
n7=NODE(width,height,depth)
n8=NODE(width,height,0)
NSEL,S,NODE,,n
NSEL,A,NODE,,n2
NSEL,A,NODE,,n3
NSEL,A,NODE,,n4
NSEL,A,NODE,,n5
NSEL,A,NODE,,n6
NSEL,A,NODE,,n7
NSEL,A,NODE,,n8
*IF,NY(n)/1000,EQ,NY(n4)/1000,THEN           ! If +ve x-face exists!
 *GET,cpnum,ACTIVE,,CP
 CP,cpnum+1,ROTX,ALL
 CP,cpnum+2,ROTY,ALL
 CP,cpnum+3,ROTZ,ALL
 *IF,n5,EQ,nfixy,THEN
  D,ALL,UX
 *ELSEIF,n6,EQ,nfixy,THEN
  D,ALL,UX
 *ELSEIF,n7,EQ,nfixy,THEN
  D,ALL,UX
 *ELSEIF,n8,EQ,nfixy,THEN
  D,ALL,UX
 *ELSE
  *GET,cpnum,ACTIVE,,CP
  CP,cpnum+1,UX,ALL
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n3,EQ,nfixx,THEN
  D,n,UZ
  D,n5,UZ
  D,n6,UZ
  CE,cenum+1,0,n4,UZ,1,n3,UZ,-1              ! Couple UZ
  CE,cenum+2,0,n7,UZ,1,n3,UZ,-1              ! Couple UZ
  CE,cenum+3,0,n8,UZ,1,n3,UZ,-1              ! Couple UZ
 *ELSEIF,n4,EQ,nfixx,THEN
  D,n2,UZ
  D,n5,UZ
  D,n6,UZ
  CE,cenum+1,0,n3,UZ,1,n4,UZ,-1              ! Couple UZ
  CE,cenum+2,0,n7,UZ,1,n4,UZ,-1              ! Couple UZ
  CE,cenum+3,0,n8,UZ,1,n4,UZ,-1              ! Couple UZ
 *ELSEIF,n7,EQ,nfixx,THEN
  D,n,UZ
  D,n2,UZ
```

```
  D,n5,UZ
  CE,cenum+1,0,n3,UZ,1,n7,UZ,-1                      ! Couple UZ
  CE,cenum+2,0,n4,UZ,1,n7,UZ,-1                      ! Couple UZ
  CE,cenum+3,0,n8,UZ,1,n7,UZ,-1                      ! Couple UZ
 *ELSEIF,n8,EQ,nfixx,THEN
  D,n,UZ
  D,n2,UZ
  D,n6,UZ
  CE,cenum+1,0,n3,UZ,1,n8,UZ,-1                      ! Couple UZ
  CE,cenum+2,0,n4,UZ,1,n8,UZ,-1                      ! Couple UZ
  CE,cenum+3,0,n7,UZ,1,n8,UZ,-1                      ! Couple UZ
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n3,UZ,1,n2,UZ,-1,nfixx,UZ,-1          ! Constraint
  CE,cenum+2,0,n4,UZ,1,n,UZ,-1,nfixx,UZ,-1          ! Constraint
  CE,cenum+3,0,n7,UZ,1,n6,UZ,-1,nfixx,UZ,-1          ! Constraint
  CE,cenum+4,0,n8,UZ,1,n5,UZ,-1,nfixx,UZ,-1          ! Constraint
 *ENDIF
 *GET,cenum,ACTIVE,,CE
 *IF,n2,EQ,nfixz,THEN
  D,n4,UY
  D,n5,UY
  D,n8,UY
  CE,cenum+1,0,n3,UY,1,n2,UY,-1                      ! Couple UY
  CE,cenum+2,0,n6,UY,1,n2,UY,-1                      ! Couple UY
  CE,cenum+3,0,n7,UY,1,n2,UY,-1                      ! Couple UY
 *ELSEIF,n3,EQ,nfixz,THEN
  D,n,UY
  D,n5,UY
  D,n8,UY
  CE,cenum+1,0,n2,UY,1,n3,UY,-1                      ! Couple UY
  CE,cenum+2,0,n6,UY,1,n3,UY,-1                      ! Couple UY
  CE,cenum+3,0,n7,UY,1,n3,UY,-1                      ! Couple UY
 *ELSEIF,n6,EQ,nfixz,THEN
  D,n,UY
  D,n4,UY
  D,n8,UY
  CE,cenum+1,0,n2,UY,1,n6,UY,-1                      ! Couple UY
  CE,cenum+2,0,n3,UY,1,n6,UY,-1                      ! Couple UY
  CE,cenum+3,0,n7,UY,1,n6,UY,-1                      ! Couple UY
 *ELSEIF,n7,EQ,nfixz,THEN
  D,n,UY
  D,n4,UY
  D,n5,UY
  CE,cenum+1,0,n2,UY,1,n7,UY,-1                      ! Couple UY
  CE,cenum+2,0,n3,UY,1,n7,UY,-1                      ! Couple UY
  CE,cenum+3,0,n6,UY,1,n7,UY,-1                      ! Couple UY
 *ELSE
  *GET,cenum,ACTIVE,,CE
  CE,cenum+1,0,n2,UY,1,n,UY,-1,nfixz,UY,-1          ! Constraint
  CE,cenum+2,0,n3,UY,1,n4,UY,-1,nfixz,UY,-1          ! Constraint
  CE,cenum+3,0,n6,UY,1,n5,UY,-1,nfixz,UY,-1          ! Constraint
  CE,cenum+4,0,n7,UY,1,n8,UY,-1,nfixz,UY,-1          ! Constraint
 *ENDIF
*ELSE                                                ! If no +ve x-face
exists
!!!!!!!!!!
*ENDIF
*ENDIF
NSEL,ALL
!C***finished corners

/EOF
```

3. ANSYS macro – mgstn

```
! Macro for generating model and mesh of the 'Statistical Model' of wood
□
!   - no right hand face created
□

□
!   height = cell height
□
!   aratio = wall area ratio
!   thick = total wall thickness
!   theta = honeycomb angle
!   mu = microfibril angle

/com, ET set to i
ET,  i,SHELL91
□
KEYOPT,i,3,9                    ! Creates 9-layered element
□

/com, ET set to i

□
ti=thick
R,1                             ! Set element layer thicknesses and
orient.
RMOD,1,1,3,70,0.045*thick,0,0,0                        ! S3
RMOD,1,7,2,-mu,0.3*thick,0,0,0                      ! S2
RMOD,1,13,1,70,0.0325*thick,0,0,0                  ! S1
RMOD,1,19,1,-70,0.0325*thick,0,0,0                 ! S1
RMOD,1,25,4,0,0.18*thick+t1,0,0,0                    ! M+P
RMOD,1,31,1,70,0.0325*t1,0,0,0                  ! S1
RMOD,1,37,1,-70,0.0325*t1,0,0,0                 ! S1
RMOD,1,43,2,mu,0.3*t1,0,0,0                      ! S2
RMOD,1,49,3,-70,0.045*t1,0,0,0                       ! S3


MP,EX,   1,47.813               ! S1 layer props
MP,EY,   1,8.547                !  12% moisture content
MP,EZ,   1,8.104
MP,NUXY,1,0.092
MP,NUYZ,1,0.370
MP,NUXZ,1,0.041
MP,GXY ,1,4.804
MP,GYZ ,1,2.665
MP,GXZ ,1,2.657

*USE,s2mpnt

MP,EX,   3,45.399               ! S3 layer props
MP,EY,   3,8.431                !  12% moisture content
MP,EZ,   3,8.064
MP,NUXY,3,0.094
MP,NUYZ,3,0.377
MP,NUXZ,3,0.043
MP,GXY ,3,4.673
MP,GYZ ,3,2.680
MP,GXZ ,3,2.648

MP,EX,   4,9.551               ! M+P layer props
MP,EY,   4,9.551               !  12% moisture content
```

```
MP,EZ,   4,5.437
MP,NUXY,4,0.301
MP,NUYZ,4,0.169
MP,NUXZ,4,0.169
MP,GXY ,4,3.670
MP,GYZ ,4,1.829
MP,GXZ ,4,1.829

*AFUN,DEG                       ! Radians to degrees

h=height-width*tan(theta)
depth=height/2                 ! depth = model depth

CSYS,0
K,6001,0,0,0                      ! Create model
K,6002,0,h/2,0
K,6003,-width/2,height/2,0
K,6004,-width/2,height/2+h,0
K,6005,0,0,depth
!k,6,0,height+h,0

L,6001,6002
□
L,6002,6003
□
L,6003,6004
□
L,6001,6005
□

□
ADRAG,P50X
□
 3
 1
1
□
2
□
3
□
4
□

□

NUMMRG,ALL

ESHAPE,2
ESIZE,,1                        ! Determines mesh size

LOCAL,12,0,,,,,,90              ! Orientates the element co-ord sys for
ESYS,12                         !  consistent layer angle definitions

AMESH,1,3                       ! Generates mesh

CSYS,0
ARSYM,X,2
CLOCA,11,0,width/2,(height+h)/2,0
ARSYM,Y,1,2
```

```
ARSYM,Y,4
□

□
CSYS,0
□

□
NUMMRG,ALL
□
NUMCMP,ALL
□
/com, clear mgstn
```

4. ANSYS macro - s2mpt

```
! Library of s2 material properties at mu=0 to mu=50, normal
distribution
!  12% Moisture Content

*set,S2(0,0),        0,      5,     10,     15,     20,     25,     30,     35,
40,     45,     50 !assign MFA to column 0, as index
*set,S2(0,1),1,63.961,63.838,63.468,62.858,62.014,60.950,59.686,58.252,5
6.683,55.022,53.313 !assign ex, column 1
*set,S2(0,2),2, 9.848, 9.843, 9.830, 9.812, 9.764, 9.783, 9.787, 9.817,
9.880, 9.987,10.146 !assign ey, column 2
*set,S2(0,3),3, 9.158, 9.159, 9.164, 9.812, 9.182, 9.197, 9.128, 9.244,
9.276, 9.315, 9.361 !assign ez, column 3
*set,S2(0,4),4, 0.051, 0.051, 0.054, 0.057, 0.063, 0.069, 0.077, 0.086,
0.095, 0.105, 0.116 !assign nuxy, column 4
*set,S2(0,5),5, 0.389, 0.389, 0.389, 0.388, 0.387, 0.386, 0.383, 0.379,
0.374, 0.368, 0.360 !assign nuyz, column 5
*set,S2(0,6),6, 0.047, 0.047, 0.046, 0.045, 0.044, 0.042, 0.041, 0.039,
0.037, 0.036, 0.035 !assign nuxz, column 6
*set,S2(0,7),7, 3.385, 3.431, 3.569, 3.791, 4.090, 4.454, 4.871, 5.324,
5.800, 6.284, 6.763 !assign gxy, column 7
*set,S2(0,8),8, 2.962, 2.962, 2.962, 2.962, 2.963, 2.963, 2.964, 2.964,
2.965, 2.965, 2.966 !assign gyz, column 8
*set,S2(0,9),9, 3.016, 3.016, 3.016, 3.016, 3.015, 3.015, 3.014, 3.014,
3.013, 3.013, 3.012 !assign gxz, column 9

MP,EX,2,S2(mu,1)
MP,EY,2,S2(mu,2)
MP,EZ,2,S2(mu,3)
MP,NUXY,2,S2(mu,4)
MP,NUYZ,2,S2(mu,5)
MP,NUXZ,2,S2(mu,6)
MP,GXY,2,S2(mu,7)
MP,GYZ,2,S2(mu,8)
MP,GXZ,2,S2(mu,9)
```

5. ANSYS macro - gwring(xx).mac (growth ring model)

```
! smear version of gwring macros
□
!/MENU,OFF
□
!/NOPR
□

□
!geometry information
□
*ASK,remark,Enter remark enclosed in single quotes,'Nil'
□
*ask,iwave,average width (radial diameter),34.28                    !width
average
*ask,iheight,average height,26.53
*ask,imfave,average MFA,32.86                                  !MFA average
*ask,idenave,average density,449.98                            !density
average
*ask,itheta,hexgon angle,15.886
*ask,cwidth,width of growth ring in um,1885


*DIM,t,,4
*DIM,we,,4
*DIM,c,,4,6,6                              ! Compliance matrix
*DIM,s,,4,6,6                              ! Stiffness matrix
*DIM,savg,,6,6                             ! Final stiffness matrix
*DIM,cavg,,6,6                             ! Final compliance matrix
*DIM,id,,6                                 ! Column of identity matrix
*DIM,alphat,,4
*DIM,sdum,,6,6
*DIM,cdum,,6,6

/PREP7
! cyclic data for stain cell method
 *dim,S2,table,10,11          !array definition for ULIB S2mpnt


 t(1)=-0.861136311
t(2)=-0.339981044
t(3)=-t(2)
t(4)=-t(1)
we(1)=0.347854845
we(2)=0.652145155
we(3)=we(2)
we(4)=we(1)

!width data (radial)
 *dim,w,table,7,2          !width array
 *set,w(0,0),0,0, 0.8,   0.84, .85,  0.86, 1      !x co-ord
 *set,w(0,1),1,1.117,    .877, .7,    .7,  1.159,      1.117 !y co-ord

!mfa values, xray cell, average=33.2
  *dim,mfa,table,6,2
  *set,mfa(0,0),0,       0,     0.8,   0.84, 0.86, 1      !x co-ord
  *set,mfa(0,1),1,       1.0788,       0.92, 0.84, 1.0961,     1.0788

  !density trial
```

```
  *dim,d,table,6,2
  *set,d(0,0),0,  0,     0.58, 0.84, 1.          !x co-ord
  *set,d(0,1),1,  .72,   .9646,       1.58, .72

!data from growth ring 14,CSIRO
/COM, DATA

*do,i1,1,4

!intialisation for next alpha value
dist=0.                                                        ! start
distance from growth ring
deltax=0.
wold=0.
cuma=0
afactor=1
i=1
dist=0      !normalised distance
cw=0  !cumulative cell width
cden=0      !cumulative density
cmu=0 !cumulative mfa
wave=iwave
height=iheight
mfave=imfave
denave=idenave
theta=itheta

width=w(dist,1)*wave                    ! width  = cell width
W1=WIDTH
cw=cw+width                   !cell 1 width
cent=width/2
dist=cent/cwidth
den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460

cden=cden+den
cmu=cmu+mu



alpha=0.5*(1+t(i1))                ! 4-pt Gauss-Legendre point

/COM, SUBROUTINE GWRING1
*USE,gwring1                             ! Custom macro for model
generation

! define origin of entire growth ring
LOCAL,13,0,-width/2*(1+alpha),0,0,0,0    ! Co-ord. system at corner of
model

!calculating line length of cell
*get,lmin,line,,num,min
*GET,LEN1,LINE,1,LENG
*GET,LEN2,LINE,2,LENG
*GET,LEN3,LINE,3,LENG
*GET,LEN4,LINE,4,LENG
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA
```

```
csys,0
/com,! generating other elements using ATRAN and ARSCALE

*do,i,2,100

width=w(dist,1)*wave
cw=cw+width

*if,cw,gt,cwidth,then
/com, cell width exceeded
*exit
*endif


cent=cw-width/2
dist= cent/ cwidth

den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460
cden=cden+den
cmu=cmu+mu

csys,13
*GET,ANUM,AREA,,NUM,MAX                 !GET AREA START NUMBER
*get,xmax,node,,mxloc,x                 !GET START OF NEW CELL CO-ORD, FROM
CSYS13

CLOCAL,14,0,xmax,0,0                    ! Co-ord system for start of new
cell
csys,13

ATRAN,14,1,7,,,1,0                      !ATRAN to start of next cell
SCALE=Width/w1
CSYS,14
*GET,ANUM,AREA,,NUM,MAX                 !GET AREA START NUMBER
ASEL,S,AREA,,ANUM-6,ANUM,,,0            !SELECT NEW AREA FOR ELEMENT
GENERATION
ARSCALE,ANUM-6,ANUM,,SCALE,,,,1,1
*use,gwring1


!calculating line length of cell
X2N=X2*SCALE
X3N=X3*SCALE
X4N=X4*SCALE
LEN2=SQRT((Y2-Y3)**2+(X2N-X3N)**2)
LEN4=SQRT((Y4-Y2)**2+(X4N-X2N)**2)
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA


!define new cell element type
ASEL,ALL
csys,13

*ENDDO


cw=cw-width
```

```
i=i-1         !number of cell
swave=cw/i
sden=cden/i
smu=cmu/i

*GET,height,KP,,MXLOC,Y
*GET,width,KP,,MXLOC,X


cuma=cuma/width/height



nummrg,all
numcmp,all


/com, starting of FEM calculation

NSEL,S,LOC,X,width                        ! Get points where forces act
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!
NSEL,S,LOC,X,width*0.9999,width*1.0001     ! Get points where forces act
NSEL,U,LOC,X,0                            !  *0.001 avoids default *0.005
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height*0.9999,height*1.0001
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
NSEL,U,LOC,X,0                            ! In case no +ve X-face exists
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!!!!!!!

/COM, USING CNSTRNTS
*ULIB,CNSTRNTS
/com, passed cnstrnts
/COM, USING NOR_CPLE
*USE,NOR_CPLE                       ! Apply normal load constraints
/COM, PASSED NOR_CPLE
WSORT
FINISH

/SOLU

NSEL,ALL
F,nfixx,FX,1                        ! x-normal loading
SOLVE

FDELE,ALL
F,nfixy,FY,1                        ! y-normal loading
SOLVE

FDELE,ALL
F,nfixz,FZ,1                        ! z-normal loading
```

```
SOLVE

FINISH

!!!!!!!!!!!!!!!!!!!! SHEAR STRESS !!!!!!!!!!!!!!!!!

/PREP7
LSCLEAR,ALL
CPDELE,ALL                    ! Clear previous loads and constraints
CEDELE,ALL
*USE,YX_CPLE                  ! Apply yx-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FX,1                  ! yx-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
CPDELE,ALL                    ! Clear previous loads and constraints
CEDELE,ALL
*USE,YZ_CPLE                  ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FZ,1                  ! yz-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
CPDELE,ALL                    ! Clear previous loads and constraints
CEDELE,ALL
*USE,XZ_CPLE                  ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixx,FZ,1                  ! xz-shear loading
ANTYPE,,REST
SOLVE
FINISH


/POST1

SET,1                                 ! Calculates moduli
EX=width/(UX(nfixx)*height*depth)
NUXY=-UY(nfixy)*width/(UX(nfixx)*height)
NUXZ=-UZ(nfixz)*width/(UX(nfixx)*depth)
SET,2
EY=height/(UY(nfixy)*width*depth)
NUYX=-UX(nfixx)*height/(UY(nfixy)*width)
NUYZ=-UZ(nfixz)*height/(UY(nfixy)*depth)
SET,3
EZ=depth/(UZ(nfixz)*height*width)
NUZX=-UX(nfixx)*depth/(UZ(nfixz)*width)
NUZY=-UY(nfixy)*depth/(UZ(nfixz)*height)
SET,4
```

```
GYX=height/(UX(nfixy)*width*depth)
SET,5
GYZ=height/(UZ(nfixy)*width*depth)
SET,6
GXZ=width/(UZ(nfixx)*height*depth)
c(i1,1,1)=1/ex
c(i1,1,2)=-nuyx/ey
c(i1,1,3)=-nuzx/ez
c(i1,2,1)=-nuxy/ex
c(i1,2,2)=1/ey
c(i1,2,3)=-nuzy/ez
c(i1,3,1)=-nuxz/ex
c(i1,3,2)=-nuyz/ey
c(i1,3,3)=1/ez
c(i1,4,4)=1/gyz
c(i1,5,5)=1/gxz
c(i1,6,6)=1/gyx


/PREP7
*IF,i1,NE,4,THEN

FDELE,ALL
DDELE,ALL
CPDELE,ALL
CEDELE,ALL
ACLEAR,ALL
ADELE,ALL
LDELE,ALL
KDELE,ALL

*ENDIF
*ENDDO

*DO,i,1,4                    ! s=inverse(c)
 *DO,j,1,6
  *DO,k,1,6                  ! Initialise id vector for given j
   *IF,k,EQ,j,THEN
    id(k)=1
   *ELSE
    id(k)=0
   *ENDIF
  *ENDDO
  *DO,l,1,6
   *DO,m,1,6
    cdum(l,m)=c(i,l,m)
   *ENDDO
  *ENDDO
  *MOPER,sdum(1,j),cdum(1,1),SOLV,id(1)
  *DO,l,1,6
   s(i,l,j)=sdum(l,j)
  *ENDDO
 *ENDDO
*ENDDO

*DO,j,1,6
 *DO,k,1,6
  savg(j,k)=0
  *DO,i,1,4
   savg(j,k)=savg(j,k)+0.5*we(i)*s(i,j,k)
  *ENDDO
```

```
 *ENDDO
*ENDDO

*DO,j,1,6                ! cavg=inverse(savg)
 *DO,k,1,6               ! Initialise id vector for given j
  *IF,k,EQ,j,THEN
   id(k)=1
  *ELSE
   id(k)=0
  *ENDIF
 *ENDDO
 *MOPER,cavg(1,j),savg(1,1),SOLV,id(1)
*ENDDO

ex=1/cavg(1,1)
ey=1/cavg(2,2)
ez=1/cavg(3,3)
nuxy=-cavg(2,1)*ex
nuyx=-cavg(1,2)*ey
nuxz=-cavg(3,1)*ex
nuzx=-cavg(1,3)*ez
nuzy=-cavg(2,3)*ez
nuyz=-cavg(3,2)*ey
gyz=1/cavg(4,4)
gxz=1/cavg(5,5)
gyx=1/cavg(6,6)


/OUTPUT,gwxx,txt,,APPEND
aave=denave/1460

*vwrite
('')
*vwrite
('***********************gw(xx).mac,gauss int***************')
*vwrite,remark
(1A32)
*vwrite,wave,mfave,denave
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)
*vwrite,cwidth,aave
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of read in data*****')


*VWRITE
('')
*vwrite
('******calculated data******')
*VWRITE,i
('cell number=',f8.3)
*vwrite,swave,smu,sden
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)

*vwrite,cw,cuma
('growth ring width=',f8.3,'    area ratio=',f8.3)
```

```
*vwrite
('*****end of calculated data*****')
*vwrite
('ex,ey,ez,nuyx,nuzx,nuxy,gyx,gyz,gxz')
*vwrite,ex,ey,ez,nuyx,nuzx,nuxy,gyx,gyz,gxz     ! Maxm of 10 outputs
(9E11.4)
*vwrite
('nuzy,nuxz,nuyz ')
*vwrite,nuzy,nuxz,nuyz    ! Maxm of 10 outputs
(3E11.4)
/OUTPUT,TERM

FINISH
```

6. ANSYS macro - gw(xray).mac (Xray MFA model)

```
! Ansys macro to analyse the 'statistical model' of wood with
!  wall offset, alpha =0.65, entire growth ring, modified from
! STOFF.mac, with periodic growth ring- 18 April 97

!/MENU,OFF
/NOPR

!geometry information
*ask,wave,average width (radial diameter),34.2845                  !width
average
*ask,height,average height,26.53
*ask,mfave,average MFA,33.2                                   !MFA average
*ask,denave,average density,449.98                            !density
average
*ask,alpha,offset factor-alpha,0.65
*ask,theta,hexgon angle,15.886
*ask,cwidth,width of growth ring in um,1885
dist=0.                                                       ! start
distance from growth ring
deltax=0.
wold=0.
cuma=0
afactor=1
i=1
dist=0      !normalised distance
cw=0  !cumulative cell width
cden=0      !cumulative density
cmu=0 !cumulative mfa


/PREP7
! cyclic data for stain cell method
 *dim,S2,table,10,11          !array definition for ULIB S2mpnt

!width data (radial)
 *dim,w,table,6,2        !width array
 *set,w(0,0),0,0, 0.75, 0.85,  .884,      1,          !x co-ord
 *set,w(0,1),1,1.12,     .9,   .72,  1.1072,1.12      !y co-ord

 !mfa values
, xray cell, average=33.2
  *dim,mfa,table,6,2
  *set,mfa(0,0),0,      0,     0.8,   0.84, 0.86, 1     !x co-ord
  *set,mfa(0,1),1,      1.0788,       0.92, 0.84, 1.0961,     1.0788



  *dim,d,table,6,2
  *set,d(0,0),0,  0,    0.6,  0.85, 0.884,      1           !x co-ord
  *set,d(0,1),1,  .78,  .9789,      1.5524,     0.8745      ,.78


!data from growth ring 14,CSIRO
/COM, DATA


width=w(dist,1)*wave                    ! width  = cell width
W1=WIDTH
```

```
cw=cw+width                     !cell 1 width
cent=width/2
dist=cent/cwidth
den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460

cden=cden+den
cmu=cmu+mu


/COM, SUBROUTINE GWRING1
*USE,gwring1                               ! Custom macro for model
generation

! define origin of entire growth ring
LOCAL,13,0,-width/2*(1+alpha),0,0,0,0    ! Co-ord. system at corner of
model

!calculating line length of cell
*get,lmin,line,,num,min
*GET,LEN1,LINE,1,LENG
*GET,LEN2,LINE,2,LENG
*GET,LEN3,LINE,3,LENG
*GET,LEN4,LINE,4,LENG
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA

csys,0
/com,! generating other elements using ATRAN and ARSCALE

*do,i,2,100

width=w(dist,1)*wave
cw=cw+width

*if,cw,gt,cwidth,then
/com, cell width exceeded
*exit
*endif


cent=cw-width/2
dist= cent/ cwidth

den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460
cden=cden+den
cmu=cmu+mu

csys,13
*GET,ANUM,AREA,,NUM,MAX               !GET AREA START NUMBER
*get,xmax,node,,mxloc,x               !GET START OF NEW CELL CO-ORD, FROM
CSYS13

CLOCAL,14,0,xmax,0,0                  ! Co-ord system for start of new
cell
csys,13
```

```
ATRAN,14,1,7,,,1,0                    !ATRAN to start of next cell
SCALE=Width/w1
CSYS,14
*GET,ANUM,AREA,,NUM,MAX               !GET AREA START NUMBER
ASEL,S,AREA,,ANUM-6,ANUM,,,0          !SELECT NEW AREA FOR ELEMENT
GENERATION
ARSCALE,ANUM-6,ANUM,,SCALE,,,,1,1
*use,gwring1


!calculating line length of cell
X2N=X2*SCALE
X3N=X3*SCALE
X4N=X4*SCALE
LEN2=SQRT((Y2-Y3)**2+(X2N-X3N)**2)
LEN4=SQRT((Y4-Y2)**2+(X4N-X2N)**2)
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA


!define new cell element type
ASEL,ALL
csys,13

*ENDDO


cw=cw-width
i=i-1        !number of cell
swave=cw/i
sden=cden/i
smu=cmu/i

*GET,height,KP,,MXLOC,Y
*GET,width,KP,,MXLOC,X


cuma=cuma/width/height



nummrg,all
numcmp,all


/com, starting of FEM calculation

NSEL,S,LOC,X,width                        ! Get points where forces act
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!
NSEL,S,LOC,X,width*0.999,width*1.001      ! Get points where forces act
NSEL,U,LOC,X,0                            !  *0.001 avoids default *0.005
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height*0.999,height*1.001
```

```
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
NSEL,U,LOC,X,0                          ! In case no +ve X-face exists
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!!!!!!!

/COM, USING CNSTRNTS
*ULIB,CNSTRNTS
/com, passed cnstrnts
/COM, USING NOR_CPLE
*USE,NOR_CPLE                     ! Apply normal load constraints
/COM, PASSED NOR_CPLE
WSORT
FINISH

/SOLU

NSEL,ALL
F,nfixx,FX,1                      ! x-normal loading
SOLVE

FDELE,ALL
F,nfixy,FY,1                      ! y-normal loading
SOLVE

FDELE,ALL
F,nfixz,FZ,1                      ! z-normal loading
SOLVE

FINISH

!!!!!!!!!!!!!!!!!!!!! SHEAR STRESS !!!!!!!!!!!!!!!!!

/PREP7
LSCLEAR,ALL
CPDELE,ALL                       ! Clear previous loads and constraints
CEDELE,ALL
*USE,YX_CPLE                     ! Apply yx-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FX,1                     ! yx-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
CPDELE,ALL                       ! Clear previous loads and constraints
CEDELE,ALL
*USE,YZ_CPLE                     ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FZ,1                     ! yz-shear loading
ANTYPE,,REST
SOLVE
FINISH
```

```
/PREP7
LSCLEAR,ALL
CPDELE,ALL                        ! Clear previous loads and constraints
CEDELE,ALL
*USE,XZ_CPLE                      ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixx,FZ,1                      ! xz-shear loading
ANTYPE,,REST
SOLVE
FINISH


/POST1

SET,1                                     ! Calculates moduli
EX=width/(UX(nfixx)*height*depth)
NUXY=-UY(nfixy)*width/(UX(nfixx)*height)
NUXZ=-UZ(nfixz)*width/(UX(nfixx)*depth)
SET,2
EY=height/(UY(nfixy)*width*depth)
NUYX=-UX(nfixx)*height/(UY(nfixy)*width)
NUYZ=-UZ(nfixz)*height/(UY(nfixy)*depth)
SET,3
EZ=depth/(UZ(nfixz)*height*width)
NUZX=-UX(nfixx)*depth/(UZ(nfixz)*width)
NUZY=-UY(nfixy)*depth/(UZ(nfixz)*height)
SET,4
GYX=height/(UX(nfixy)*width*depth)
SET,5
GYZ=height/(UZ(nfixy)*width*depth)
SET,6
GXZ=width/(UZ(nfixx)*height*depth)

/OUTPUT,gwstain,txt,,APPEND
aave=denave/1460

*vwrite
('')
*vwrite
('**********************gw(x-ray).mac********************')
*vwrite
('Growth ring 14 analysis, read in data')
*vwrite,wave,mfave,denave
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)
*vwrite,cwidth,aave
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of read in data*****')


*VWRITE
('')
*vwrite
('******calculated data******')
*VWRITE,i
('cell number=',f8.3)
*vwrite,swave,smu,sden
```

```
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)

*vwrite,cw,cuma
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of calculated data*****')

*VWRITE
('ER          ET        EL       VTR             VLR         VLT             GTR
GTL       GRL')
*vwrite,ex,ey,ez,nuyx,nuzx,nuzy,gyx,gyz,gxz    ! Maxm of 10 outputs
(9E11.4)
/OUTPUT,TERM

FINISH
```

7. ANSYS macro - gw(stain2).mac (Stained-cell MFA model)

```
! Ansys macro to analyse the 'statistical model' of wood with
□
!  wall offset, alpha =0.65, entire growth ring, modified from
! STOFF.mac, with periodic growth ring- 18 April 97

!/MENU,OFF
/NOPR

!geometry information
*ask,wave,average width (radial diameter),34.28                !width
average
*ask,height,average height,26.53
*ask,mfave,average MFA,32.86                                   !MFA average
*ask,denave,average density,449.98                             !density
average
*ask,alpha,offset factor-alpha,0.65
*ask,theta,hexgon angle,15.886
*ask,cwidth,width of growth ring in um,1885
dist=0.                                                        ! start
distance from growth ring
deltax=0.
wold=0.
cuma=0
afactor=1
i=1
dist=0      !normalised distance
cw=0  !cumulative cell width
cden=0      !cumulative density
cmu=0 !cumulative mfa



/PREP7
! cyclic data for stain cell method
 *dim,S2,table,10,11          !array definition for ULIB S2mpnt

!width data (radial)
 *dim,w,table,6,2          !width array
 *set,w(0,0),0,0, 0.75, 0.85,  .884,      1,          !x co-ord
 *set,w(0,1),1,1.12,    .9,   .72,  1.1072,1.12       !y co-ord

 !mfa values, stain cell, AVERAGE= 32.86
  *dim,mfa,table,6,2
  *set,mfa(0,0),0,      0,     0.55, 0.84, 1.    !x co-ord
  *set,mfa(0,1),1,      1.05, 1.05, .83,  1.05


   !density trial

  *dim,d,table,6,2
  *set,d(0,0),0,  0,     0.6,  0.85, 0.884,      1            !x co-ord
  *set,d(0,1),1,  .78,  .9789,      1.5524,      0.8745       ,.78

!data from growth ring 14,CSIRO
/COM, DATA


width=w(dist,1)*wave                    ! width  = cell width
W1=WIDTH
```

```
cw=cw+width                         !cell 1 width
cent=width/2
dist=cent/cwidth
den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460

cden=cden+den
cmu=cmu+mu


/COM, SUBROUTINE GWRING1
*USE,gwring1                                   ! Custom macro for model
generation

! define origin of entire growth ring
LOCAL,13,0,-width/2*(1+alpha),0,0,0,0    ! Co-ord. system at corner of
model

!calculating line length of cell
*get,lmin,line,,num,min
*GET,LEN1,LINE,1,LENG
*GET,LEN2,LINE,2,LENG
*GET,LEN3,LINE,3,LENG
*GET,LEN4,LINE,4,LENG
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA

csys,0
/com,! generating other elements using ATRAN and ARSCALE

*do,i,2,100

width=w(dist,1)*wave
cw=cw+width

*if,cw,gt,cwidth,then
/com, cell width exceeded
*exit
*endif


cent=cw-width/2
dist= cent/ cwidth

den=d(dist,1)*denave
mu=mfa(dist,1)*mfave
aratior=den/1460
cden=cden+den
cmu=cmu+mu

csys,13
*GET,ANUM,AREA,,NUM,MAX               !GET AREA START NUMBER
*get,xmax,node,,mxloc,x               !GET START OF NEW CELL CO-ORD, FROM
CSYS13

CLOCAL,14,0,xmax,0,0                  ! Co-ord system for start of new
cell
csys,13
```

```
ATRAN,14,1,7,,,1,0                        !ATRAN to start of next cell
SCALE=Width/w1
CSYS,14
*GET,ANUM,AREA,,NUM,MAX                   !GET AREA START NUMBER
ASEL,S,AREA,,ANUM-6,ANUM,,,0              !SELECT NEW AREA FOR ELEMENT
GENERATION
ARSCALE,ANUM-6,ANUM,,SCALE,,,,1,1
*use,gwring1


!calculating line length of cell
X2N=X2*SCALE
X3N=X3*SCALE
X4N=X4*SCALE
LEN2=SQRT((Y2-Y3)**2+(X2N-X3N)**2)
LEN4=SQRT((Y4-Y2)**2+(X4N-X2N)**2)
LENG1=(LEN1+LEN2+LEN3/2+LEN4)*2
CELAREA=LENG1*THICK
CUMA=CUMA+CELAREA


!define new cell element type
ASEL,ALL
csys,13

*ENDDO


cw=cw-width
i=i-1         !number of cell
swave=cw/i
sden=cden/i
smu=cmu/i

*GET,height,KP,,MXLOC,Y
*GET,width,KP,,MXLOC,X


cuma=cuma/width/height



nummrg,all
numcmp,all


/com, starting of FEM calculation

NSEL,S,LOC,X,width                           ! Get points where forces act
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!
NSEL,S,LOC,X,width*0.999,width*1.001      ! Get points where forces act
NSEL,U,LOC,X,0                            !  *0.001 avoids default *0.005
*GET,nfixx,NODE,,NUM,MIN
NSEL,S,LOC,Y,height*0.999,height*1.001
```

```
*GET,nfixy,NODE,,NUM,MIN
NSEL,S,LOC,Z,depth
NSEL,U,LOC,X,0                           ! In case no +ve X-face exists
*GET,nfixz,NODE,,NUM,MIN
NSEL,ALL

!!!!!!!!!!!!!!!!!!!!! DIRECT STRESS !!!!!!!!!!!!!!!!!!!!!!!

/COM, USING CNSTRNTS
*ULIB,CNSTRNTS
/com, passed cnstrnts
/COM, USING NOR_CPLE
*USE,NOR_CPLE                     ! Apply normal load constraints
/COM, PASSED NOR_CPLE
WSORT
FINISH

/SOLU

NSEL,ALL
F,nfixx,FX,1                      ! x-normal loading
SOLVE

FDELE,ALL
F,nfixy,FY,1                      ! y-normal loading
SOLVE

FDELE,ALL
F,nfixz,FZ,1                      ! z-normal loading
SOLVE

FINISH

!!!!!!!!!!!!!!!!!!!! SHEAR STRESS !!!!!!!!!!!!!!!!

/PREP7
LSCLEAR,ALL
CPDELE,ALL                       ! Clear previous loads and constraints
CEDELE,ALL
*USE,YX_CPLE                     ! Apply yx-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FX,1                     ! yx-shear loading
ANTYPE,,REST
SOLVE
FINISH

/PREP7
LSCLEAR,ALL
CPDELE,ALL                       ! Clear previous loads and constraints
CEDELE,ALL
*USE,YZ_CPLE                     ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixy,FZ,1                     ! yz-shear loading
ANTYPE,,REST
SOLVE
FINISH
```

```
/PREP7
LSCLEAR,ALL
CPDELE,ALL                      ! Clear previous loads and constraints
CEDELE,ALL
*USE,XZ_CPLE                    ! Apply yz-shear constraints
FINISH
/SOLU
NSEL,ALL
F,nfixx,FZ,1                    ! xz-shear loading
ANTYPE,,REST
SOLVE
FINISH

/POST1

SET,1                                    ! Calculates moduli
EX=width/(UX(nfixx)*height*depth)
NUXY=-UY(nfixy)*width/(UX(nfixx)*height)
NUXZ=-UZ(nfixz)*width/(UX(nfixx)*depth)
SET,2
EY=height/(UY(nfixy)*width*depth)
NUYX=-UX(nfixx)*height/(UY(nfixy)*width)
NUYZ=-UZ(nfixz)*height/(UY(nfixy)*depth)
SET,3
EZ=depth/(UZ(nfixz)*height*width)
NUZX=-UX(nfixx)*depth/(UZ(nfixz)*width)
NUZY=-UY(nfixy)*depth/(UZ(nfixz)*height)
SET,4
GYX=height/(UX(nfixy)*width*depth)
SET,5
GYZ=height/(UZ(nfixy)*width*depth)
SET,6
GXZ=width/(UZ(nfixx)*height*depth)

/OUTPUT,gwstain,txt,,APPEND
aave=denave/1460

*vwrite
('')
*vwrite
('***********************gw(stain2).mac********************')
*vwrite
('Growth ring 14 analysis, read in data')
*vwrite,wave,mfave,denave
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)
*vwrite,cwidth,aave
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of read in data*****')


*VWRITE
('')
*vwrite
('******calculated data******')
*VWRITE,i
('cell number=',f8.3)
```

```
*vwrite,swave,smu,sden
('width ave=',f10.3,'   MFA ave=',f10.3,'   density ave=',f10.3)
*vwrite,height,alpha
('height ave=',f10.3,'   alpha=',f10.3)

*vwrite,cw,cuma
('growth ring width=',f8.1,'    area ratio=',f8.3)
*vwrite
('*****end of calculated data*****')

*VWRITE
('ER         ET        EL       VTR             VLR          VLT              GTR
GTL        GRL')
*vwrite,ex,ey,ez,nuyx,nuzx,nuzy,gyx,gyz,gxz    ! Maxm of 10 outputs
(9E11.4)
/OUTPUT,TERM

FINISH
```

8. Analytical solution, theta=0,

Et, theta=0, non-normalised

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
Eo	0.434	0.4382	0.451	0.4752	0.5136
Ep	0.434	0.438	0.451	0.475	0.514
FEM	0.427	0.4303	0.4422	0.4643	0.4997
Gib(s)	0.444	0.444	0.444	0.444	0.444
Gib(a)	0.434	0.434	0.433	0.433	0.433

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
Eo	3.251	3.27	3.35	3.51	3.776
Ep	3.251	3.274	3.363	3.529	3.797
FEM	3.213	3.23	3.309	3.46	3.703
Gib(s)	3.556	3.556	3.556	3.556	3.556
Gib(a)	3.251	3.246	3.24	3.234	3.228

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
Eo	20.699	20.61	20.89	21.59	22.81
Ep	20.699	20.69	21.07	21.87	23.2
FEM	20.699	20.67	20.99	21.67	22.81
Gib(s)	28.448	28.448	28.448	28.448	28.448
Gib(a)	20.699	20.58	20.46	20.34	20.23

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
Eo	133.09	129.56	128.09	128.61	131.28
Ep	133.09	130.95	130.8	132.64	136.62
FEM	133.09	130.55	128.05	130.73	132.58
Gib(s)	444.5	444.5	444.5	444.5	444.5
Gib(a)	133.09	131.13	129.222	127.37	125.57

Et, theta=0, normalised

nu=0

t/H	0	0.025	0.05	0.1	0.25
Eo	1	0.977477	0.914229	0.727608	0.299415
Ep	1	0.977477	0.914229	0.727608	0.299415
FEM	1	0.961712	0.903543	0.727608	0.299415
Gib(s)	1	1	1	1	1
Gib(a)	1	0.977477	0.914229	0.727608	0.299415

nu=0.1

t/H	0	0.025	0.05	0.1	0.25
Eo	1	0.986937	0.919573	0.72448	0.291474
Ep	1	0.986486	0.920697	0.727292	0.294601
FEM	1	0.969144	0.908324	0.726589	0.293701
Gib(s)	1	1	1	1	1
Gib(a)	1	0.977477	0.912823	0.723425	0.295006

nu=0.2

t/H	0	0.025	0.05	0.1	0.25
Eo		1.015766	0.94207	0.734322	0.288166
Ep		1.015766	0.945726	0.74065	0.294263
FEM		0.995946	0.93054	0.737837	0.288076
Gib(s)	1	1	1	1	1
Gib(a)	1	0.975225	0.911136	0.719207	0.290713

nu=0.3

t/H	0	0.025	0.05	0.1	0.25
Eo		1.07027	0.987064	0.758929	0.289336
Ep		1.06982	0.992407	0.763771	0.298403
FEM		1.045721	0.973003	0.761741	0.294106
Gib(s)	1	1	1	1	1
Gib(a)	1	0.975225	0.909449	0.714989	0.286547

nu=0.4

t/H	0	0.025	0.05	0.1	0.25
Eo		1.156757	1.061867	0.801814	0.295343
Ep		1.157658	1.067773	0.815523	0.307357
FEM		1.12545	1.041339	0.801814	0.298268
Gib(s)	1	1	1	1	1
Gib(a)	1	0.975225	0.907762	0.711122	0.282497

Grt, theta=0, non-normalised

values x E-4

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
Go	0.1479	0.1494	0.154	0.1623	0.1758
Gp	0.1479	0.1494	0.154	0.1624	0.1758
FEM	0.147	0.1483	0.1528	0.1609	0.175
Gib(s)	0.1483	0.1483	0.1483	0.1483	0.1483
Gib(a)	0.1479	0.1479	0.1478	0.1478	0.1477

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
Go	1.173	1.184	1.219	1.284	1.387
Gp	1.174	1.184	1.219	1.284	1.387
FEM	1.166	1.178	1.212	1.275	1.377
Gib(s)	1.187	1.187	1.187	1.187	1.187
Gib(a)	1.174	1.172	1.171	1.17	1.169

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
Go	9.09	9.14	9.373	9.816	10.53
Gp	9.09	9.14	9.373	9.817	10.54
FEM	9.075	9.14	9.373	9.817	10.54
Gib(s)	9.492	9.492	9.492	9.492	9.492
Gib(a)	9.09	9.051	9.016	8.98	8.944

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
Go	11.62	11.47	11.5	11.71	12.134
Gp	11.62	11.47	11.5	11.72	12.144
FEM	11.61	11.47	11.49	11.7	12.14
Gib(s)	14.83	14.83	14.83	14.83	14.83
Gib(a)	11.62	11.38	11.15	10.93	10.72

Grt, theta=0, normalised

Grt x 01-4

nu=0

t/H=0.025	0	0.025	0.05	0.1	0.25
Go	1	0.997303	0.988206	0.957649	0.783547
Gp	1	0.997303	0.989048	0.957649	0.783547
FEM	1	0.991234	0.982308	0.956068	0.782873
Gib(s)	1	1	1	1	1
Gib(a)	1	0.997303	0.989048	0.957649	0.783547

nu=0.1

t/H	0	0.025	0.05	0.1	0.25
Go		1.007417	0.997473	0.962916	0.773432
Gp		1.007417	0.997473	0.962916	0.773432
FEM		1	0.992418	0.962916	0.773432
Gib(s)	1	1	1	1	1
Gib(a)	1	0.997303	0.987363	0.95354	0.767363

nu=0.2

t/H	0	0.025	0.05	0.1	0.25
Go		1.038436	1.026959	0.987463	0.775455
Gp		1.038436	1.026959	0.987463	0.775455
FEM		1.030344	1.021061	0.987463	0.774781
Gib(s)	1	1	1	1	1
Gib(a)	1	0.996628	0.986521	0.949853	0.751854

nu=0.3

t/H	0	0.025	0.05	0.1	0.25
Go		1.094403	1.081719	1.034134	0.789616
Gp		1.095078	1.081719	1.034239	0.79029
FEM		1.084963	1.074136	1.034239	0.788941
Gib(s)	1	1	1	1	1
Gib(a)	1	0.996628	0.985678	0.94606	0.73702

nu=0.4

t/H	0	0.025	0.05	0.1	0.25
Go		1.185435	1.168492	1.109355	0.818206
Gp		1.185435	1.168492	1.110409	0.818881
FEM		1.18004	1.160067	1.110409	0.818611
Gib(s)	1	1	1	1	1
Gib(a)	1	0.995954	0.984836	0.942267	0.722859

9. Analytical solution, theta=15.

Er, non-normalised

t/H=0.025					
nu	0	0.1	0.2	0.3	0.4
Eo	1.207	1.217	1.252	1.317	1.423
Ep	1.207	1.218	1.255	1.322	1.43
FEM	1.188	1.197	1.23	1.291	1.388
Esg	1.268	1.268	1.268	1.268	1.268
Eag	1.207	1.206	1.206	1.205	1.205

t/H=0.05					
nu	0	0.1	0.2	0.3	0.4
Eo	8.432	8.460	8.658	9.055	9.708
Ep	8.432	8.493	8.729	9.168	9.874
FEM	8.345	8.389	8.589	8.967	9.57
Es	10.14	10.14	10.14	10.14	10.14
Ea	8.432	8.419	8.406	8.393	8.38

t/H=0.1					
nu	0	0.1	0.2	0.3	0.4
Eo	44.777	44.43	44.925	46.34	48.9
Ep	44.777	44.9	45.89	47.85	51.063
FEM	44.786	44.82	45.51	46.93	49.22
Es	81.151	81.151	81.151	81.151	81.151
Ea	44.777	44.59	44.41	44.23	44.05

t/H=0.25					
nu	0	0.1	0.2	0.3	0.4
Eo	208.65	204	202.9	205.6	212.7
Ep	208.65	207.9	210.9	217.8	229.6
FEM	208.67	207.1	207.7	210.3	215.1
Es	1267.99	1267.99	1268	1268	1268
Ea	208.65	207.1	205.5	203.9	202.4

Er, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Eo	0.95189	0.83156	0.55177	0.16455
Ep	0.95189	0.83156	0.55177	0.16455
FEM	0.93691	0.82298	0.55188	0.16457
Gib(s)	1	1	1	1
Gib(a)	0.95189	0.83156	0.55177	0.16455

nu=0.1

t/H	0.025	0.05	0.1	0.25
Eo	0.95978	0.83432	0.5475	0.16088
Ep	0.96057	0.83757	0.55329	0.16396
FEM	0.94401	0.82732	0.5523	0.16333
Gib(s)	1	1	1	1
Gib(a)	0.9511	0.83028	0.54947	0.16333

nu=0.2

t/H	0.025	0.05	0.1	0.25
Eo	0.98738	0.85385	0.5536	0.16002
Ep	0.98975	0.86085	0.56549	0.16632
FEM	0.97003	0.84704	0.56081	0.1638
Gib(s)	1	1	1	1
Gib(a)	0.9511	0.82899	0.54725	0.16207

nu=0.3

t/H	0.025	0.05	0.1	0.25
Eo	1.03864	0.893	0.57103	0.16215
Ep	1.04259	0.90414	0.58964	0.17177
FEM	1.01814	0.88432	0.5783	0.16585
Gib(s)	1	1	1	1
Gib(a)	0.95032	0.82771	0.54503	0.1608

nu=0.4

t/H	0.025	0.05	0.1	0.25
Eo	1.12224	0.9574	0.60258	0.16774
Ep	1.12776	0.97377	0.62923	0.18107
FEM	1.09464	0.94379	0.60652	0.16964
Gib(s)	1	1	1	1
Gib(a)	0.95032	0.82643	0.54282	0.15962

Et, non-normalised

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
Eo	0.3722	0.3756	0.388	0.4074	0.4404
Ep	0.3722	0.3757	0.387	0.4078	0.441
FEM	0.366	0.3692	0.3794	0.3985	0.4291
Gib(s)	0.379	0.379	0.379	0.379	0.3793
Gib(a)	0.3722	0.3721	0.3719	0.3717	0.3716

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
Eo	2.819	2.836	2.909	3.050	3.279
Ep	2.819	2.839	2.917	3.062	3.295
FEM	2.787	2.803	2.872	3.005	3.218
Gib(s)	3.035	3.035	3.035	3.034	3.034
Gib(a)	2.819	2.814	2.81	2.805	2.8

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
Eo	18.595	18.52	18.788	19.43	20.53
Ep	18.595	18.59	18.94	19.66	20.86
FEM	18.603	18.59	18.889	19.51	20.56
Gib(s)	24.278	24.278	24.278	24.278	24.278
Gib(a)	18.595	18.49	18.38	18.28	18.18

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
Eo	130.35	126.65	124.92	125.09	127.27
Ep	130.35	128	127.6	129.1	132.52
FEM	130.39	127.8	126.4	126.4	127.6
Gib(s)	379.35	379.35	379.35	379.35	379.35
Gib(a)	130.35	128.3	126.3	124.4	122.5

Et, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Eo	0.982058	0.92883	0.76592	0.343614
Ep	0.982058	0.92883	0.76592	0.343614
FEM	0.965699	0.918287	0.766249	0.34372
Gib(s)	1	1	1	1
Gib(a)	0.982058	0.92883	0.76592	0.343614

nu=0.1

t/H	0.025	0.05	0.1	0.25
Eo	0.991029	0.934432	0.762831	0.333861
Ep	0.991293	0.93542	0.765714	0.337419
FEM	0.974142	0.923558	0.765714	0.336892
Gib(s)	1	1	1	1
Gib(a)	0.981794	0.927183	0.761595	0.33821

nu=0.2

t/H	0.025	0.05	0.1	0.25
Eo	1.023747	0.958484	0.773869	0.3293
Ep	1.021108	0.96112	0.78013	0.336365
FEM	1.001055	0.946293	0.778029	0.333202
Gib(s)	1	1	1	1
Gib(a)	0.981266	0.925865	0.757064	0.332938

nu=0.3

t/H	0.025	0.05	0.1	0.25
Eo	1.074934	1.005274	0.800313	0.329748
Ep	1.075989	1.009229	0.809787	0.340319
FEM	1.051451	0.990442	0.803608	0.333202
Gib(s)	1	1	1	1
Gib(a)	0.980739	0.924522	0.752945	0.327929

nu=0.4

t/H	0.025	0.05	0.1	0.25
Eo	1.161086	1.080751	0.845622	0.335495
Ep	1.162668	1.086025	0.859214	0.349334
FEM	1.131294	1.060646	0.846857	0.336365
Gib(s)	1	1	1	1
Gib(a)	0.979699	0.922874	0.748826	0.322921

Grt, non-normalised

values x E-4

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	0.271	0.274	0.282	0.298	0.322
ps	0.271	0.274	0.283	0.298	0.322
FEM	0.269	0.272	0.280	0.295	0.319
Gib(s)	0.273	0.273	0.273	0.273	0.272
Gib(a)	0.271	0.271	0.271	0.271	0.271

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	2.144	2.162	2.226	2.341	2.528
ps	2.144	2.162	2.226	2.342	2.529
FEM	2.129	2.150	2.211	2.324	2.506
Gib(s)	2.180	2.180	2.180	2.180	2.180
Gib(a)	2.144	2.141	2.138	2.135	2.132

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	16.350	16.410	16.790	17.540	18.760
ps	16.350	16.410	16.800	17.550	18.780
FEM	16.330	16.420	16.800	17.550	18.780
Gib(s)	17.440	17.440	17.440	17.440	17.440
Gib(a)	16.340	16.260	16.170	16.080	16.000

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	192.080	188.400	187.600	189.500	194.400
pser	192.100	188.600	188.000	190.100	195.200
FEM	192.100	188.600	188.000	190.100	195.200
Gib(s)	272.500	272.500	272.540	272.540	272.540
Gib(a)	192.100	187.300	182.700	178.300	174.100

Grt, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
keo	0.99267	0.98349	0.9375	0.70488
ps	0.99267	0.98349	0.9375	0.70495
FEM	0.98535	0.97661	0.93635	0.70495
Gib(s)	1	1	1	1
Gib(a)	0.99267	0.98349	0.93693	0.70495

nu=0.1

t/H	0.025	0.05	0.1	0.25
keo	1.00366	0.99174	0.94094	0.69138
ps	1.00366	0.99174	0.94094	0.69211
FEM	0.99634	0.98624	0.94151	0.69211
Gib(s)	1	1	1	1
Gib(a)	0.99267	0.98211	0.93234	0.68734

nu=0.2

t/H	0.025	0.05	0.1	0.25
keo	1.03297	1.0211	0.96273	0.68834
ps	1.0348	1.0211	0.9633	0.68981
FEM	1.02564	1.01422	0.9633	0.68981
Gib(s)	1	1	1	1
Gib(a)	0.99341	0.98073	0.92718	0.67036

nu=0.3

t/H	0.025	0.05	0.1	0.25
keo	1.09048	1.07385	1.00573	0.69531
pser	1.09084	1.07431	1.00631	0.69751
FEM	1.08059	1.06606	1.00631	0.69751
Gib(s)	1	1	1	1
Gib(a)	0.99304	0.97936	0.92202	0.65422

nu=0.4

t/H	0.025	0.05	0.1	0.25
keo	1.18382	1.15963	1.07569	0.71329
pser	1.18382	1.16009	1.07683	0.71623
FEM	1.17279	1.14954	1.07683	0.71623
Gib(s)	1	1	1	1
Gib(a)	0.99632	0.97798	0.91743	0.63881

Vrt, non-normalised

values x
E-4

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	1.747	1.745	1.743	1.741	1.739
ps	1.747	1.748	1.748	1.749	1.751
FEM	1.749	1.748	1.7467	1.7439	1.738
Gib(s)	1.828	1.828	1.828	1.828	1.828
Gib(a)	1.747	1.748	1.748	1.748	1.748

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	1.546	1.54	1.534	1.528	1.522
ps	1.546	1.547	1.55	1.554	1.56
FEM	1.549	1.546	1.543	1.534	1.519
Gib(s)	1.828	1.828	1.828	1.828	1.828
Gib(a)	1.546	1.547	1.547	1.547	1.548

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	1.079	1.072	1.066	1.061	1.058
ps	1.079	1.0853	1.095	1.109	1.13
FEM	1.079	1.08	1.073	1.057	1.034
Gib(s)	1.828	1.828	1.828	1.828	1.828
Gib(a)	1.079	1.082	1.085	1.088	1.091

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	0.432	0.44	0.45	0.466	0.4882
pser	0.432	0.451	0.476	0.509	0.554
FEM	0.432	0.445	0.446	0.449	0.44
Gib(s)	1.828	1.828	1.828	1.828	1.828
Gib(a)	0.432	0.442	0.453	0.463	0.473

Vrt, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Vo	0.955689	0.845733	0.590263	0.236324
Vp	0.955689	0.845733	0.590263	0.236324
FEM	0.956783	0.847374	0.590263	0.236324
Gib(s)	1	1	1	1
Gib(a)	0.955689	0.845733	0.590263	0.236324

nu=0.1

t/H	0.025	0.05	0.1	0.25
Vo	0.954595	0.842451	0.586433	0.2407
Vp	0.956236	0.84628	0.593709	0.246718
FEM	0.956236	0.845733	0.59081	0.243435
Gib(s)	1	1	1	1
Gib(a)	0.956236	0.84628	0.591904	0.241794

nu=0.2

t/H	0.025	0.05	0.1	0.25
Vo	0.953501	0.839168	0.583151	0.246171
Vp	0.956236	0.847921	0.599015	0.260394
FEM	0.955525	0.844092	0.58698	0.243982
Gib(s)	1	1	1	1
Gib(a)	0.956236	0.84628	0.593545	0.247812

nu=0.3

t/H	0.025	0.05	0.1	0.25
Vo	0.952407	0.835886	0.580416	0.254923
Vp	0.956783	0.850109	0.606674	0.278446
FEM	0.953993	0.839168	0.578228	0.245624
Gib(s)	1	1	1	1
Gib(a)	0.956236	0.84628	0.595186	0.253282

nu=0.4

t/H	0.025	0.05	0.1	0.25
Vo	0.951313	0.832604	0.578775	0.267068
Vp	0.957877	0.853392	0.618162	0.303063
FEM	0.950766	0.830963	0.565646	0.2407
Gib(s)	1	1	1	1
Gib(a)	0.956236	0.846827	0.596827	0.258753

10. Analytical solution, theta=30

Er, non-normalised

theta=30

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	0.2848	0.2874	0.2961	0.312	0.3374
pser	0.2848	0.2875	0.2962	0.3122	0.337
FEM	0.281	0.283	0.2908	0.3056	0.329
Esg	0.2887	0.2887	0.2887	0.2887	0.2887
Eag	0.2848	0.2846	0.2845	0.2843	0.2842

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	2.191	2.205	2.265	2.377	2.558
pser	2.191	2.208	2.269	2.384	2.569
FEM	2.167	2.18	2.236	2.343	2.514
Es	2.309	2.309	2.309	2.309	2.309
Ea	2.191	2.188	2.185	2.182	2.179

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	15.193	15.162	15.414	15.98	16.94
pser	15.1931	15.213	15.52	16.15	17.183
FEM	15.19	15.2	15.48	16.05	16.98
Es	18.475	18.475	18.475	18.475	18.475
Ea	15.193	15.119	15.045	14.972	14.899

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	122.84	119.38	117.76	117.91	119.94
pser	122.84	120.65	120.28	121.67	124.96
FEM	122.8	120.4	119.3	119.4	120.8
Es	288.6	288.6	288.6	288.6	288.6
Ea	122.84	120.911	119.041	117.229	115.47

Er, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Eo	0.986491	0.948896	0.822355	0.425641
Ep	0.986491	0.948896	0.82236	0.425641
FEM	0.973329	0.938502	0.822192	0.425502
Gib(s)	1	1	1	1
Gib(a)	0.986491	0.948896	0.822355	0.425641

nu=0.1

t/H	0.025	0.05	0.1	0.25
Eo	0.995497	0.954959	0.820677	0.413652
Ep	0.995843	0.956258	0.823437	0.418053
FEM	0.980256	0.944132	0.822733	0.417186
Gib(s)	1	1	1	1
Gib(a)	0.985798	0.947596	0.818349	0.418957

nu=0.2

t/H	0.025	0.05	0.1	0.25
Eo	1.025632	0.980944	0.834317	0.408039
Ep	1.025979	0.982676	0.840054	0.416771
FEM	1.007274	0.968385	0.837889	0.413375
Gib(s)	1	1	1	1
Gib(a)	0.985452	0.946297	0.814344	0.412477

nu=0.3

t/H	0.025	0.05	0.1	0.25
Eo	1.080707	1.02945	0.864953	0.408559
Ep	1.081399	1.032482	0.874154	0.421587
FEM	1.058538	1.014725	0.868742	0.413721
Gib(s)	1	1	1	1
Gib(a)	0.984759	0.944998	0.810392	0.406199

nu=0.4

t/H	0.025	0.05	0.1	0.25
Eo	1.168687	1.107839	0.916915	0.415593
Ep	1.167302	1.112603	0.930068	0.432987
FEM	1.139591	1.088783	0.91908	0.418572
Gib(s)	1	1	1	1
Gib(a)	0.984413	0.943699	0.806441	0.400104

Et, non-normalised

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	0.2848	0.2874	0.2961	0.312	0.3373
ps	0.2848	0.2875	0.2963	0.3122	0.3377
FEM	0.281	0.283	0.2908	0.3056	0.329
Gib(s)	0.2886	0.2886	0.2886	0.2886	0.2886
Gib(a)	0.28482	0.2847	0.2846	0.2845	0.2844

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	2.191	2.205	2.265	2.377	2.558
ps	2.191	2.208	2.269	2.385	2.569
FEM	2.167	2.18	2.236	2.343	2.514
Gib(s)	2.309	2.309	2.309	2.309	2.309
Gib(a)	2.191	2.188	2.1849	2.182	2.179

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	15.19	15.162	15.414	15.98	16.939
ps	15.193	15.21	15.52	16.15	17.183
FEM	15.19	15.2	15.48	16.05	16.98
Gib(s)	18.48	18.48	18.48	18.48	18.48
Gib(a)	15.193	15.12	15.04	14.97	14.9

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	122.84	119.381	117.76	117.907	119.94
pser	122.84	120.7	120.28	121.7	124.96
FEM	122.8	120.4	119.3	119.4	120.8
Gib(s)	288.7	288.7	288.7	288.7	288.7
Gib(a)	122.84	120.9	119.04	117.2	115.5

Et, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Eo	0.986833	0.948896	0.82197	0.425494
Ep	0.986833	0.948896	0.822132	0.425494
FEM	0.973666	0.938502	0.82197	0.425355
Gib(s)	1	1	1	1
Gib(a)	0.986902	0.948896	0.822132	0.425494

nu=0.1

t/H	0.025	0.05	0.1	0.25
Eo	0.995842	0.954959	0.820455	0.413512
Ep	0.996188	0.956258	0.823052	0.418081
FEM	0.980596	0.944132	0.822511	0.417042
Gib(s)	1	1	1	1
Gib(a)	0.986486	0.947596	0.818182	0.418774

nu=0.2

t/H	0.025	0.05	0.1	0.25
Eo	1.025988	0.980944	0.834091	0.407897
Ep	1.026681	0.982676	0.839827	0.416626
FEM	1.007623	0.968385	0.837662	0.413232
Gib(s)	1	1	1	1
Gib(a)	0.98614	0.946254	0.813853	0.412331

nu=0.3

t/H	0.025	0.05	0.1	0.25
Eo	1.081081	1.02945	0.864719	0.408407
Ep	1.081774	1.032915	0.873918	0.421545
FEM	1.058905	1.014725	0.868506	0.413578
Gib(s)	1	1	1	1
Gib(a)	0.985793	0.944998	0.810065	0.405958

nu=0.4

t/H	0.025	0.05	0.1	0.25
Eo	1.168746	1.107839	0.916613	0.415449
Ep	1.170132	1.112603	0.929816	0.432837
FEM	1.139986	1.088783	0.918831	0.418427
Gib(s)	1	1	1	1
Gib(a)	0.985447	0.943699	0.806277	0.400069

Grt, non-normalised

values x E-4

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	0.716	0.722	0.7442	0.7842	0.8482
ps	0.716	0.7223	0.7443	0.7843	0.8483
FEM	0.707	0.713	0.7331	0.7709	0.8316
Gib(s)	0.721	0.721	0.721	0.721	0.721
Gib(a)	0.716	0.715	0.714	0.714	0.7139

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	5.584	5.624	5.779	6.069	6.536
ps	5.584	5.626	5.783	6.075	6.546
FEM	5.534	5.57	5.718	5.999	6.45
Gib(s)	5.773	5.773	5.773	5.773	5.773
Gib(a)	5.584	5.571	5.557	5.545	5.532

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	40.658	40.64	41.37	42.96	45.59
ps	40.658	40.69	41.48	43.118	45.82
FEM	40.658	40.69	41.48	43.12	45.82
Gib(s)	46.188	46.19	46.19	46.19	46.187
Gib(a)	40.658	40.32	39.98	39.653	39.31

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	390.101	378.633	372.55	371.47	375.43
pser	390.101	380.336	375.88	376.36	381.85
FEM	390.102	380.3	375.9	376.4	381.8
Gib(s)	721.69	721.69	721.69	721.69	721.69
Gib(a)	390.101	377.85	366.338	355.51	345.305

Grt, normalised

nu=0

t/H	0.025	0.05	0.1	0.25
Go	0.993065	0.967261	0.880272	0.540538
Gp	0.993065	0.967261	0.880272	0.540538
FEM	0.980583	0.9586	0.880272	0.54054
Gib(s)	1	1	1	1
Gib(a)	0.993065	0.967261	0.880272	0.540538

nu=0.1

t/H	0.025	0.05	0.1	0.25
Go	1.001387	0.97419	0.879844	0.524648
Gp	1.001803	0.974537	0.880927	0.527007
FEM	0.988904	0.964836	0.880927	0.526958
Gib(s)	1	1	1	1
Gib(a)	0.991678	0.96501	0.872916	0.523563

nu=0.2

t/H	0.025	0.05	0.1	0.25
Go	1.032178	1.001039	0.895648	0.516219
Gp	1.032316	1.001732	0.89803	0.520833
FEM	1.016782	0.990473	0.89803	0.520861
Gib(s)	1	1	1	1
Gib(a)	0.990291	0.962584	0.865555	0.507611

nu=0.3

t/H	0.025	0.05	0.1	0.25
Go	1.087656	1.051273	0.930071	0.514722
Gp	1.087795	1.052312	0.933492	0.521498
FEM	1.069209	1.039148	0.933535	0.521554
Gib(s)	1	1	1	1
Gib(a)	0.990291	0.960506	0.858476	0.492608

nu=0.4

t/H	0.025	0.05	0.1	0.25
Go	1.176422	1.132167	0.987074	0.52021
Gp	1.17656	1.133899	0.992054	0.529105
FEM	1.153398	1.11727	0.992054	0.529036
Gib(s)	1	1	1	1
Gib(a)	0.990153	0.958254	0.851105	0.478467

Vrt, non-normalised

t/H=0.025

nu	0	0.1	0.2	0.3	0.4
keo	0.9901	0.99	0.99	0.99	0.9902
ps	0.9901	0.9902	0.9904	0.9909	0.9909
FEM	0.99	0.9902	0.99	0.9895	0.9886
Gib(s)	1	1	1	1	1
Gib(a)	0.9901	0.9901	0.9901	0.9901	0.9901

t/H=0.05

nu	0	0.1	0.2	0.3	0.4
keo	0.9621	0.9611	0.9607	0.9611	0.9627
ps	0.9621	0.9624	0.9634	0.9652	0.9684
FEM	0.962	0.9623	0.9613	0.9595	0.9565
Gib(s)	1	1	1	1	1
Gib(a)	0.9621	0.9621	0.9621	0.9622	0.9622

t/H=0.1

nu	0	0.1	0.2	0.3	0.4
keo	0.8684	0.8663	0.866	0.869	0.875
ps	0.8684	0.8703	0.8744	0.8814	0.8929
FEM	0.868	0.8684	0.866	0.861	0.853
Gib(s)	1	1	1	1	1
Gib(a)	0.8684	0.869	0.8697	0.8703	0.871

t/H=0.25

nu	0	0.1	0.2	0.3	0.4
keo	0.5745	0.578	0.5873	0.6043	0.617
pser	0.5745	0.5878	0.6073	0.6352	0.6757
FEM	0.574	0.5829	0.5868	0.5864	0.5816
Gib(s)	1	1	1	1	1
Gib(a)	0.5745	0.5811	0.5876	0.5939	0.6

Vrt, normalised

nu=0				
t/H	0.025	0.05	0.1	0.25
Vo	0.9901	0.9621	0.8684	0.5745
Vp	0.9901	0.9621	0.8684	0.5745
FEM	0.99	0.962	0.868	0.574
Gib(s)	1	1	1	1
Gib(a)	0.9901	0.9621	0.8684	0.5745

nu=0.1				
t/H	0.025	0.05	0.1	0.25
Vo	0.99	0.9611	0.8663	0.578
Vp	0.9902	0.9624	0.8703	0.5878
FEM	0.9902	0.9623	0.8684	0.5829
Gib(s)	1	1	1	1
Gib(a)	0.9901	0.9621	0.869	0.5811

nu=0.2				
t/H	0.025	0.05	0.1	0.25
Vo	0.99	0.9607	0.866	0.5873
Vp	0.9904	0.9634	0.8744	0.6073
FEM	0.99	0.9613	0.866	0.5868
Gib(s)	1	1	1	1
Gib(a)	0.9901	0.9621	0.8697	0.5876

nu=0.3				
t/H	0.025	0.05	0.1	0.25
Vo	0.99	0.9611	0.869	0.6043
Vp	0.9909	0.9652	0.8814	0.6352
FEM	0.9895	0.9595	0.861	0.5864
Gib(s)	1	1	1	1
Gib(a)	0.9901	0.9622	0.8703	0.5939

nu=0.4				
t/H	0.025	0.05	0.1	0.25
Vo	0.9886	0.957	0.858	0.597
Vp	0.9909	0.9684	0.8929	0.6757
FEM	0.9886	0.9565	0.853	0.5816
Gib(s)	1	1	1	1
Gib(a)	0.9901	0.9622	0.871	0.6

11. Maple macro- Er, beam without shear (xer.ms)

```
> with(linalg):
> k:=matrix(12,12,0);
> #define kel;
> kel:=matrix(6,6,0);
> kel[1,1]:=k1;kel[1,4]:=-k1;
> kel[2,2]:=12*a1;kel[2,3]:=6*a1*L1;kel[2,5]:=-12*a1;kel[2,6]:=6*a1*L1;
> kel[3,2]:=6*a1*L1;kel[3,3]:=4*a1*L1^2;kel[3,5]:=-
6*a1*L1;kel[3,6]:=2*a1*L1^2;
> kel[4,1]:=-k1;kel[4,4]:=k1;
> kel[5,2]:=-12*a1;kel[5,3]:=-6*a1*L1;kel[5,5]:=12*a1;kel[5,6]:=-
6*a1*L1;
> kel[6,2]:=6*a1*L1;kel[6,3]:=2*a1*L1^2;kel[6,5]:=-
6*a1*L1;kel[6,6]:=4*a1*L1^2;
> #T1 for element 1 with theta
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(kel);
> # define element k2;
> ke2:=matrix(6,6,0);
> ke2[1,1]:=k2;ke2[1,4]:=-k2;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-k2;ke2[4,4]:=k2;
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> print(T2);
> ke3:=matrix(6,6,0);
> print (ke2);
>
> ke3[1,1]:=k3;ke3[1,4]:=-k3;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-k3;ke3[4,4]:=k3;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> print(ke3);
> T3:=matrix(6,6,0);
```

```
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
> print(T3);
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> K1:=multiply(transpose(T1),ke1,T1);
> K2:=multiply(transpose(T2),ke2,T2);
> K3:=multiply(transpose(T3),ke3,T3);
> print (K2);
# #define a1,a2,a3;k1,k2,k3
>
a1:=E*b*(t/2)^3/12/(h/2)^3;L1:=h/2;a2:=E*b*t^3/12/l^3;L2:=l;a3:=E*b*(t/2
)^3/12/(h/2)^3;L3:=h/2;
> #define s1,c1,s2,c2,s3,c3 and L1,L2,L3;
> s3:=1;c3:=0;s1:=1;c1:=0;s2:=s;c2:=c;
> k1:=E*b*(t/2)/(h/2);
> k2:=E*b*t/l;
> k3:=E*b*(t/2)/(h/2);
# #assembly k1, n1=first node(degree of freedom #), n2=second node;
> i:=0;j:=0;n1:=7;n2:=4;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=K1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=K1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=K1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=K1[i,j];
> od;
> od;
> #assembly k2;
> i:=0;j:=0;n1:=1;n2:=4;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=K2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=K2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=K2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=K2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3;
> i:=0;j:=0;n1:=10;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=K3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=K3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=K3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=K3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
```

```
> od;
> print(k);
> #simplify global matrix
> print(K3);
> s:=sin(theta);c:=cos(theta);print (K2);
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print (k);
# #define boundary conditions
> kg:=delrows(k,12..12);
> print(kg);
> kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,8..8);kg:=delcols(kg,8..8);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
> kg:=delrows(kg,2..2);kg:=delcols(kg,2..2);
>
> print(kg);
# #define coupled nodes
> kg1:=addcol(kg,4,2,1);
> kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,2);kg:=delrows(kg3,4..4);
> print(kg);
# #define forces
> f:=matrix(4,1,0);
> f[2,1]:=sigma*b*(h+l*sin(theta));
> linsolve(kg,f);
>
> d:=matrix(4,1,0);
> d[1,1]:=delta2;d[2,1]:=delta4;d[3,1]:=delta5;d[4,1]:=delta11;
> EN:=add(multiply(2*transpose(d),kg,d),multiply(-4*transpose(d),f));
> print (EN);
> U:=multiply(2*transpose(d),kg,d);
> simplify(EN[1,1]);
> X:=EN[1,1];
> d2:=diff(X,delta2);
> d4:=diff(X,delta4);
> d5:=diff(X,delta5);
> d11:=diff(X,delta11);
> print (d4);
> eqnset:={d2, d5, d4, d11};
> varset:={delta2, delta4, delta5, delta11};
> solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> delta11*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 + sin(theta)^4);
> strainr:=delta4*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4)/(l*cos(theta));
> Er:=sigma/strainr;
> print (Er);
>
Er:=E*t^3*cos(theta)/((l^2*sin(theta)^2)*(1+(t/l)^2*cot(theta)^2)*(h+l*s
in(theta)));
> print (Er);
> Ers:=E*t^3*cos(theta)/((l^2*sin(theta)^2)*(h+l*sin(theta)));
```

```
> print(Ers);
> drt:=delta11/(h+l*sin(theta))*(2*cos(theta)^2*sin(theta)^2 +
cos(theta)^4 + sin(theta)^4);
> print (drt);
> v:=drt/strainr;
> print(v);
> vrt:=-1* cos(theta)^2*((t/l)^2-
1)/(h+l*sin(theta))/sin(theta)/(1+cot(theta)^2*(t/l)^2);
> print (vrt);
>
>
```

12. Maple macro- Et, beam without shear (xet.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=E*b*t/(h/2);ke3[1,4]:=-E*b*t/(h/2);
>
> ke3[1,1]:=E*b*t/(h/2);ke3[1,4]:=-E*b*t/(h/2);
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-E*b*t/(h/2);ke3[4,4]:=E*b*t/(h/2);
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
```

```
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
> a1:=E*b*t^3/12/l^3;L1:=l;a2:=E*b*t^3/12/l^3;L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
> s3:=1;c3:=0;a3:=E*b*t^3/12/((h/2)^3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
>
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
```

```
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
> K:=E*b*t/l;
> # d2, d7,d8, d11
> f:=matrix(4,1,0);
> f[4,1]:=-sigma*b*l*cos(theta);
> linsolve(kg,f);
```

```
>
> d:=matrix(4,1,0);
> d[1,1]:=delta2;d[2,1]:=delta7;d[3,1]:=delta8;d[4,1]:=delta11;
> EN:=add(multiply(transpose(d),kg,d),multiply(4*transpose(d),f));
> print (EN);
> U:=multiply(transpose(d),kg,d);print (U);
> simplify(EN[1,1]);
> X:=EN[1,1];
> d2:=diff(X,delta2);
> d7:=diff(X,delta7);
> d8:=diff(X,delta8);
> d11:=diff(X,delta11);
> print (d11);
> eqnset:={d2, d7, d8, d11};
> varset:={delta2, delta7, delta8, delta11};
> #solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> print (delta2);
> delta11*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 + sin(theta)^4);
> strainr:=delta11*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4)/(l*cos(theta));
> Er:=sigma/strainr;
> print (Er);
>
Er:=E*t^3*cos(theta)/((l^2*sin(theta)^2)*(1+(t/l)^2*cot(theta)^2)*(h+l*s
in(theta)));
> print (Er);
> Ers:=E*t^3*cos(theta)/((l^2*sin(theta)^2)*(h+l*sin(theta)));
> print(Ers);
> drt:=delta11/(h+l*sin(theta))*(2*cos(theta)^2*sin(theta)^2 +
cos(theta)^4 + sin(theta)^4);
> print (drt);
> v:=drt/strainr;
> print(v);
> print(k1);
> print(k);
>
```

13. Maple macro- Grt, beam without shear (xgx.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=E*b*t/(h/2);ke3[1,4]:=-E*b*t/(h/2);
>
> ke3[1,1]:=E*b*t/(h/2);ke3[1,4]:=-E*b*t/(h/2);
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-E*b*t/(h/2);ke3[4,4]:=E*b*t/(h/2);
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
```

```
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
> a1:=E*b*t^3/12/l^3;L1:=l;a2:=E*b*t^3/12/l^3;L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
> s3:=1;c3:=0;a3:=E*b*t^3/12/((h/2)^3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
```

```
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,13..14);kg:=delcols(kg,13..14);
> kg:=delrows(kg,11..11);kg:=delcols(kg,11..11);
>
> kg1:=addcol(kg,6,3,1);# 6 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> #couple 8 and 5
> kg1:=addcol(kg,5,7,1);
>  kg:=delcols(kg1,5..5);
> kg3:=addrow(kg,5,7);kg:=delrows(kg3,5..5);
>
> #couple 7 and 4
> kg1:=addcol(kg,4,5,1);kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,5);kg:=delrows(kg3,4..4);
>
> kg1:=addcol(kg,6,3,1);# 9 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> kg1:=addcol(kg,8,7,1);# 15 to 12;
> kg:=delcols(kg1,8..8);
> kg3:=addrow(kg,8,7);kg:=delrows(kg3,8..8);
>
> f:=matrix(7,1,0);
> # d1, d2,d3, d7, d8, d10, d12
> f[6,1]:=sigma*2*l*cos(theta)*b;
```

```
> print(f);
> print(kg);
>
> linsolve(kg,f);
> solutions:=";
> K:=E*b*t/l;ur:=solutions[6,1];
> shearstrainr:=(ur)/(h+l*sin(theta));
> shearstressr:=sigma;
> Grt:=shearstressr/shearstrainr;simplify(Grt);
>
> d:=matrix(7,1,0);
>
d[1,1]:=delta1;d[2,1]:=delta2;d[3,1]:=delta3;d[4,1]:=delta7;d[5,1]:=delt
a8;d[6,1]:=delta10;d[7,1]:=
> delta12;
> EN:=add(multiply(transpose(d),kg,d),multiply(-2*transpose(d),f));
> print (EN);
> U:=multiply(transpose(d),kg,d);
> simplify(EN[1,1]);
> X:=EN[1,1];
> d2:=diff(X,delta2);d1:=diff(X,delta1);
> d3:=diff(X,delta3);d7:=diff(X,delta7);
> d8:=diff(X,delta8);d10:=diff(X,delta10);
> d12:=diff(X,delta12);
> print (d12);
> eqnset:={d1, d2, d3, d7, d8, d10, d12};
> varset:={delta1 ,delta2, delta3, delta7, delta8, delta10, delta12};
> solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> print (delta12);
> ur:=delta10;
> shearstrainr:=(ur)/(h+l*sin(theta));
> shearstressr:=sigma;
> print (shearstrainr);
> Grt:=shearstressr/shearstrainr;
> print (Grt);
> Grt:= t^3*E*(h+l*sin(theta))/(h^2*l*cos(theta)*(1+2*h)
)/(1+t^2*(1+h*sin(theta))^2*2/(h^2*(2*h+1)*2*l*c
> os(theta)^2))
> ;
> print (Grt);
>
```

14. Maple program- Er, beam with shear (ser.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model; with shear effect
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=b*t^3/12;A:=b*t;
>
e1:=12*Ew*Iw/(kappa*A*G*L1^2);e2:=12*Ew*Iw/(kappa*A*G*L2^2);e3:=12*Ew*Iw
/(kappa*A*G*L3
> ^2);
> K2:=Ew*A/(h/2);K:=Ew*A/l;
> a1:=Ew*Iw/L1^3/(1+e1);L1:=l;a2:=Ew*Iw/L2^3/(1+e2);L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=Ew*Iw/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
```

```
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
```

```
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
> K:=E*b*t/l;
> # d2, d7,d8, d11
>
>
> f:=matrix(4,1,0);
> f[2,1]:=2*sigma*b*(h+l*sin(theta));
> linsolve(kg,f);
> solutions:=";
> delta11:=solutions[4,1];
>
> delta:=delta*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> strainr:=delta/(l*cos(theta));
> Er:=sigma/strainr;
> print(Er);
> kappa:=5/6;
> print (Er);
> delta11:=solutions[4,1];
> drt:=delta11/(h+l*sin(theta))*(2*cos(theta)^2*sin(theta)^2 +
cos(theta)^4 + sin(theta)^4);
> print (drt);
> denrt:=l^2*(1+(2/kappa +2*v/kappa-1)*(t/l)^2)*l*sin(theta)*cos(theta);
> #vrt:=drt/strainr;
> vrt:=denrt/(den/cos(theta));
> print(vrt);
> evalf(vrt,2);
>
den:=(h+l*sin(theta))*l^2*sin(theta)^2*(1+(2/kappa+2/kappa*v+cot(theta)^
2)*(t/l)^2);
> num:=Ew*t^3*cos(theta);
> print (num/den);
> kappa:=5/6;
> print(num/den);
> evalf(num/den,2);
>
```

15. Maple macro-Et, beam with shear (set.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
#
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=b*t^3/12;A:=b*t;
>
e1:=12*Ew*Iw/(kappa*A*G*L1^2);e2:=12*Ew*Iw/(kappa*A*G*L2^2);e3:=12*Ew*Iw
/(kappa*A*G*L3
> ^2);
> K2:=Ew*A/(h/2);K:=Ew*A/1;
> a1:=Ew*Iw/L1^3/(1+e1);L1:=1;a2:=Ew*Iw/L2^3/(1+e2);L2:=1;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=Ew*Iw/L3^3/(1+e3);L3:=h/2;
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
>
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
```

```
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
```

```
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
> K:=E*b*t/l;
> # d2, d7,d8, d11
> f:=matrix(4,1,0);
> f[4,1]:=-2*sigma*b*l*cos(theta);
> linsolve(kg,f);
> solutions:=";
> delta13:=solutions[2,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> delta11:=solutions[4,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> straint:=delta11/(h+l*sin(theta));
> print (straint);
> Et:=sigma/straint;
> print (Et);
> #simplifying Et;
>
Et:=Ew*t^3*(h+l*sin(theta))/(l^4*cos(theta)^3*(1+(2/kappa+2*v/kappa+tan(
theta)^2+2*h/l/cos(theta
> )^2)*(t/l)^2));
> kappa:=5/6;
> evalf(Et,2);
> evalf(vrt,2);
>
```

16. Maple macro- Grt, beam with shear (Sgx.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model; with shear effect
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=b*t^3/12;A:=b*t;
>
e1:=12*Ew*Iw/(kappa*A*G*L1^2);e2:=12*Ew*Iw/(kappa*A*G*L2^2);e3:=12*Ew*Iw
/(kappa*A*G*L3
> ^2);
> K2:=Ew*A/(h/2);K:=Ew*A/l;
> a1:=Ew*Iw/L1^3/(1+e1);L1:=l;a2:=Ew*Iw/L2^3/(1+e2);L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=Ew*Iw/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
```

```
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
> print(k);
> kg:=delrows(k,13..14);kg:=delcols(kg,13..14);
> kg:=delrows(kg,11..11);kg:=delcols(kg,11..11);
>
> kg1:=addcol(kg,6,3,1);# 6 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> #couple 8 and 5
> kg1:=addcol(kg,5,7,1);
>  kg:=delcols(kg1,5..5);
> kg3:=addrow(kg,5,7);kg:=delrows(kg3,5..5);
>
> #couple 7 and 4
> kg1:=addcol(kg,4,5,1);kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,5);kg:=delrows(kg3,4..4);
>
> kg1:=addcol(kg,6,3,1);# 9 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> kg1:=addcol(kg,8,7,1);# 15 to 12;
> kg:=delcols(kg1,8..8);
> kg3:=addrow(kg,8,7);kg:=delrows(kg3,8..8);
>
> f:=matrix(7,1,0);
> # d1, d2,d3, d7, d8, d10, d12
> f[6,1]:=sigma*2*l*cos(theta)*b;
> print(f);
> print(kg);
>
> linsolve(kg,f);
```

```
> solutions:='';
> ur:=solutions[6,1];
> delta1:=solutions[1,1];
> delta2:=solutions[2,1];
> delta3:=solutions[3,1];
> delta7:=solutions[4,1];
> delta8:=solutions[5,1];
> delta12:=solutions[7,1];
> delta10:=solutions[6,1];
> shearstrainr:=(ur)/(h+l*sin(theta));
> print (shearstrainr);
> shearstressr:=sigma;
> Grt:=shearstressr/shearstrainr;
> #simplify(Grt);
> num:=Ew*t^3*(h+l*sin(theta));
> den:=h^2*l*cos(theta)*(l+2*h);
>
den1:=1+(t^2/h^2/(2*h+l))*(((l+h*sin(theta))^2)/(l*cos(theta)^2)+2*h/l/k
appa*(2*l+h)*(1+v));
> den2:=den*den1;
> Grt:=num/den2;
> kappa:=5/6;evalf(Grt,2);
>
>
```

17. Maple macro- Er, plate without shear (per.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
>
>
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
```

```
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa, keo
> A:=t;D1:=Ew*t^3/(12*(1-v^2));D2:=Ew*t^3/(12*(1-
v^2));D3:=Ew*t^3/(12*(1-v^2));
>
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3;L1:=1;a2:=D2/L2^3;L2:=1;c2:=-c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3;L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
```

```
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
>
> # d2, d7,d8, d11
> f:=matrix(4,1,0);
> f[2,1]:=2*sigma*(h+l*sin(theta));
> linsolve(kg,f);
> solutions:=";
> delta:=solutions[2,1];
```

```
>
strainr:=delta/(l*cos(theta))*(cos(theta)^4+sin(theta)^4+2*cos(theta)^2*
sin(theta)^2);
> Er:=sigma/strainr;
> #kappa:=6/5;
> Er:=Ew*t^3*cos(theta)/(l^2*sin(theta)^2)/((1+t^2*cot(theta)^2/l^2)-
v^2*(1+t^2*cot(theta)^2/l^2))/(h+
> l*sin(theta));
> print(Er);
>
> delta11:=solutions[4,1];
>
> drt:=delta11/(h+l*sin(theta))*(2*cos(theta)^2*sin(theta)^2 +
cos(theta)^4 + sin(theta)^4);
> print (drt);
> vrt:=drt/strainr;
> print (vrt);
> vrt:=-l*cos(theta)^2*((t/l)^2-
1)/(h+l*sin(theta))/sin(theta)/(1+cot(theta)^2*(t/l)^2);
> print (vrt);
>
```

18. Maple macro- Et, plate without shear (pet.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa, keo
> A:=t;D1:=Ew*t^3/(12*(1-v^2));D2:=Ew*t^3/(12*(1-
v^2));D3:=Ew*t^3/(12*(1-v^2));
>
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3;L1:=1;a2:=D2/L2^3;L2:=1;c2:=-c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3;L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
```

```
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
> s:=sin(theta); c:=cos(theta);
> # d2, d7,d8, d11
> f:=matrix(4,1,0);
>
> f[4,1]:=2*sigma*l*cos(theta);
> linsolve(kg,f);
> solutions:=";
> delta13:=solutions[2,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> delta11:=solutions[4,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> straint:=delta11/(h+l*sin(theta));
> print (straint);
> Et:=sigma/straint;
> print (Et);
>
Et:=Ew*t^3*(h+l*sin(theta))/l^4/cos(theta)^3/((1+2*t^2*h/l^3/cos(theta)^
2+ t^2*tan(theta)^2/l^2)-v^
> 2*(1+2*t^2*h/l^3/cos(theta)^2+ t^2*tan(theta)^2/l^2));
```

```
> print (Et);
>
```

19. Maple macro- Grt, plate without shear (pgx.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=4*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=2*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=2*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=4*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=4*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=2*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=2*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=4*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
>
>
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=4*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=2*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
```

```
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=2*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=4*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa, keo
> A:=t;D1:=Ew*t^3/(12*(1-v^2));D2:=Ew*t^3/(12*(1-
v^2));D3:=Ew*t^3/(12*(1-v^2));
>
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/l/(1-v^2);
> a1:=D1/L1^3;L1:=l;a2:=D2/L2^3;L2:=l;c2:=-c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3;L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
```

```
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,13..14);kg:=delcols(kg,13..14);
> kg:=delrows(kg,11..11);kg:=delcols(kg,11..11);
>
> kg1:=addcol(kg,6,3,1);# 6 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> #couple 8 and 5
> kg1:=addcol(kg,5,7,1);
>  kg:=delcols(kg1,5..5);
> kg3:=addrow(kg,5,7);kg:=delrows(kg3,5..5);
>
> #couple 7 and 4
> kg1:=addcol(kg,4,5,1);kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,5);kg:=delrows(kg3,4..4);
>
> kg1:=addcol(kg,6,3,1);# 9 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> kg1:=addcol(kg,8,7,1);# 15 to 12;
```

```
> kg:=delcols(kg1,8..8);
> kg3:=addrow(kg,8,7);kg:=delrows(kg3,8..8);
>
> f:=matrix(7,1,0);
> # d1, d2,d3, d7, d8, d10, d12
> f[6,1]:=sigma*2*l*cos(theta);
> print(f);
> print(kg);
>
> linsolve(kg,f);
> solutions:='';
> ur:=solutions[6,1];
> shearstrainr:=(ur)/(h+l*sin(theta));
> shearstressr:=sigma;
> Grt:=shearstressr/shearstrainr;
> Grt:=Ew*t^3*(h+lsin(theta))/((1-
v^2)*(h^2*l*cos(theta)*(2*h+l)+t^2*(l+h*sin(theta)^2)^2/cos(theta)))
> ;
>
> Grt:= t^3*E*(h+l*sin(theta))/(h^2*l*cos(theta)*(1+2*h)
)/(1+t^2*(l+h*sin(theta))^2*2/(h^2*(2*h+l)*2*l*c
> os(theta)^2)-
v^2*(1+t^2*(l+h*sin(theta))^2*2/(h^2*(2*h+l)*2*l*cos(theta)^2)));
> print (Grt);
>
```

20. Maple macro- Er, plate with shear (pser.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using plate element with
free y surface, final versi
> on,31 July
> # 1/2 hexagonal model; with shear effect
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=t^3/12;A:=t;D1:=Ew*t^3/(12*(1-
v^2));D2:=Ew*t^3/(12*(1-v^2));D3:=Ew*t^3/(12*
> (1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=1;a2:=D2/L2^3/(1+e2);L2:=1;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
```

```
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);simplify(ke2[2,2]);print(ke2[2,2]);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
```

```
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
>
> # d2, d7,d8, d11
>
> f:=matrix(4,1,0);
>
> f[2,1]:=2*sigma*(h+l*sin(theta));
> linsolve(kg,f);
> solutions:=";
> delta11:=solutions[4,1];
>
> delta:=solutions[2,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> strainr:=delta/(l*cos(theta));
> Er:=sigma/strainr;
> Er:=Ew*t^3*cos(theta)/(h+l*sin(theta))/l^2/sin(theta)^2/(1+(2*1.2/(1-
v)+cot(theta)^2)*(t/l)^2-v^2*(1+(
> 2*1.2/(1-v)+cot(theta)^2)*(t/l)^2));
>
> theta:=30/180*3.14159;v:=0.4;Ew:=1000;h:=30;t:=1.5;l:=30;
> c:=cos(theta);s:=sin(theta);c:=cos (theta);s:=sin(theta);evalf(t/60);
> print(Er);
> kappa:=6/5;
> delta11:=solutions[4,1];
> drt:=delta11/(h+l*sin(theta))*(2*cos(theta)^2*sin(theta)^2 +
cos(theta)^4 + sin(theta)^4);
> drt:=l^2*sin(theta)*(1+((1.4+v)/(1-v))*(t/l)^2)*cos(theta)^2;
> print (drt);
>
>
vrt:=drt*l*cos(theta)/cos(theta)/(h+l*sin(theta))/l^2/sin(theta)^2/(1+(2
*1.2+2.4*v+cot(theta)^2)*(t/l)^
> 2);
> print(vrt);
> Er:=Ew*t^3*cos(theta)/
>
(h+l*sin(theta))/l^2/sin(theta)^2/(1+(2*1.2+2.4*v+cot(theta)^2)*(t/l)^2-
v^2*(1+cot(theta)^2*(t/l)^2));
> print(Er);
>
```

21. Maple macro- Et, plate with shear (pset.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using plate element with
free y surface, final versi
> on,31 July
> # 1/2 hexagonal model; with shear effect
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=t^3/12;A:=t;D1:=Ew*t^3/(12*(1-
v^2));D2:=Ew*t^3/(12*(1-v^2));D3:=Ew*t^3/(12*
> (1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=1;a2:=D2/L2^3/(1+e2);L2:=1;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
```

```
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
>
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
>
> # d2, d7,d8, d11
>
> f:=matrix(4,1,0);
> f[4,1]:=2*sigma*l*cos(theta);
```

```
> linsolve(kg,f);
> solutions:='';
> delta13:=solutions[2,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> delta11:=solutions[4,1]*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 +
sin(theta)^4);
> straint:=delta11/(h+l*sin(theta));
> print (straint);
> Et:=sigma/straint;
> print (Et);
> Et:=Ew*t^3*(h+l*sin(theta))/l^4/cos(theta)^3/(1-
v^2)/(1+(tan(theta)^2+2/kappa/(1-v)+2*h/l/cos(theta
> )^2)*(t/l)^2);
> print(Et);
>
>
Et:=Ew*t^3*(h+l*sin(theta))/(l^4*cos(theta)^3*(1+(2*1.2+2*v*1.2+tan(thet
a)^2+2*h/l/cos(theta)^2)*(t
> /l)^2-v^2*(1+(tan(theta)^2+2*h/l/cos(theta)^2)*(t/l)^2)));
> print(Et);
>
```

22. Maple macro- Grt, plate with shear (psgx.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model; with shear effect
>
> k:=matrix(15,15,0);
>
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K;ke1[1,4]:=-K;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K;ke1[4,4]:=K;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K;ke2[1,4]:=-K;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K;ke2[4,4]:=K;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2;ke3[1,4]:=-K2;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2;ke3[4,4]:=K2;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
```

```
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=t^3/12;A:=t;D1:=Ew*t^3/(12*(1-
v^2));D2:=Ew*t^3/(12*(1-v^2));D3:=Ew*t^3/(12*
> (1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=l;a2:=D2/L2^3/(1+e2);L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
```

```
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
> print(k);
> kg:=delrows(k,13..14);kg:=delcols(kg,13..14);
> kg:=delrows(kg,11..11);kg:=delcols(kg,11..11);
>
> kg1:=addcol(kg,6,3,1);# 6 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> #couple 8 and 5
> kg1:=addcol(kg,5,7,1);
>  kg:=delcols(kg1,5..5);
> kg3:=addrow(kg,5,7);kg:=delrows(kg3,5..5);
>
> #couple 7 and 4
> kg1:=addcol(kg,4,5,1);kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,5);kg:=delrows(kg3,4..4);
>
> kg1:=addcol(kg,6,3,1);# 9 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> kg1:=addcol(kg,8,7,1);# 15 to 12;
> kg:=delcols(kg1,8..8);
> kg3:=addrow(kg,8,7);kg:=delrows(kg3,8..8);
>
> f:=matrix(7,1,0);
> # d1, d2,d3, d7, d8, d10, d12
> f[6,1]:=sigma*2*l*cos(theta);
> print(f);
> print(kg);
```

```
>
> linsolve(kg,f);
> solutions:=";
> ur:=solutions[6,1];
> delta1:=solutions[1,1];
> delta2:=solutions[2,1];
> delta3:=solutions[3,1];
> delta7:=solutions[4,1];
> delta8:=solutions[5,1];
> delta12:=solutions[7,1];
> delta10:=solutions[6,1];
> shearstrainr:=(ur)/(h+l*sin(theta));
> print (shearstrainr);
> shearstressr:=sigma;
> Grt:=shearstressr/shearstrainr;
>
Grt:=Ew*t^3*(h+l*sin(theta))/h^2/l/c/(l+2*h)/(1+t^2*(((((l+h*s)^2/(l*c^2
)+h*2.4*(h+2*l)*(1+v)/(l)))/(h^2)/
> (l+2*h)))-
v^2*(1+t^2*(l+h*s)^2/(h^2)/(l+2*h)/l/cos(theta)^2));c:=cos(theta);s:=sin
(theta);
> print (Grt);
>
>
```

23. Maple macro- Er, plate with shear and eo strain (psereo.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using plate element with
free y surface, final versi
> on,31 July
> # 1/2 hexagonal model; with shear effect
# # calculation of keo
# # macro for calculating keo, strain energy for strain in y direction
due to plate element, K is modified with keo
> keo:=v*Ew*t/(2*l+h)/(v^2-1);
>
>
> k:=matrix(15,15,0);
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K+keo;ke1[1,4]:=-K-keo;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K-keo;ke1[4,4]:=K+keo;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K+keo;ke2[1,4]:=-K-keo;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K-keo;ke2[4,4]:=K+keo;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2+keo;ke3[1,4]:=-K2-keo;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
```

```
> ke3[4,1]:=-K2-keo;ke3[4,4]:=K2+keo;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa, keo
> G:=Ew/2/(1+v);A:=t;D1:=Ew*t^3/(12*(1-v^2));D2:=Ew*t^3/(12*(1-
v^2));D3:=Ew*t^3/(12*(1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/l/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=l;a2:=D2/L2^3/(1+e2);L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
```

```
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
>
> # d2, d7,d8, d11
> f:=matrix(4,1,0);
> f[2,1]:=2*sigma*(h+l*sin(theta));
>
```

```
# # differentiation of energy equation
> d:=matrix(4,1,0);
> d[1,1]:=delta2;d[2,1]:=delta7;d[3,1]:=delta8;d[4,1]:=delta11;
> EN:=add(multiply(2*transpose(d),kg,d),multiply(-4*transpose(d),f));
> print (EN);
> U:=multiply(2*transpose(d),kg,d);
>
# # including eo^2 terms
# #energy due to eo
> ENeo:=ky*h/2*(delta8/(2*l+h))^2+ky*h/2*((delta2-
delta11)/(2*l+h))^2+ky*l*((s2*delta2+c2*delta7-s2*
> delta8)/(2*l+h))^2-ky*l*((-s1*delta2+c1*delta7+s1*delta8)/(2*l+h))^2;
> ky:=1/2*Ew*t/(1-v^2);
> X:=EN[1,1]+2*ENeo;
>
> d2:=diff(X,delta2);
> d7:=diff(X,delta7);
> d8:=diff(X,delta8);
> d11:=diff(X,delta11);
> print (d7);
> eqnset:={d2, d7, d8, d11};
> varset:={delta2, delta7, delta8, delta11};
> solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> delta11*(2*cos(theta)^2*sin(theta)^2 + cos(theta)^4 + sin(theta)^4);
> strainr:=delta7/(l*cos(theta));
> Er:=sigma/strainr;
> print (Er);
> drt:=delta11/(h+l*sin(theta));print(Er);
>
>
#theta:=30/180*3.14159;v:=0.4;Ew:=1000;h:=30;t:=1.5;l:=30;kappa:=5/6;eva
lf (t/60);
> #c:=cos(theta);s:=sin(theta);c:=cos (theta);s:=sin(theta);
> print (drt);
> vrt:=drt/strainr;
> vrt1:=l*(1+(2.4/(1-v)-(2*l+h)/(h+2*l-
v*l))*(t/l)^2)*cos(theta)^2/sin(theta)/(h+l*sin(theta))/(1+(2.4+2.4*v+
> cot(theta)^2*(2*l+h)/(h+l*2-v*l))*(t/l)^2);
> print (vrt1);
>
>
Ers:=Ew*t^3*c/((h+l*s)*l^2*s^2)/(1+(2.4+2.4*v+cot(theta)^2*(2*l+h)/(2*l+
h-v*l))*(t/l)^2-v^2*(1+cot(th
> eta)^2*(2*l+h)/(2*l+h-v*l)*(t/l)^2));evalf(Ers); evalf (Er);
> print (Er);
>
```

24. Maple macro- Et, plate with shear and eo strain (pseteo.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using plate element with
free y surface, final versi
> on,31 July
> # 1/2 hexagonal model; with shear effect
>
> keo:=v*Ew*t/(2*l+h)/(v^2-1);
>
>
> k:=matrix(15,15,0);
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K+keo;ke1[1,4]:=-K-keo;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K-keo;ke1[4,4]:=K+keo;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K+keo;ke2[1,4]:=-K-keo;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K-keo;ke2[4,4]:=K+keo;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2+keo;ke3[1,4]:=-K2-keo;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2-keo;ke3[4,4]:=K2+keo;
```

```
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=t^3/12;A:=t;D1:=Ew*t^3/(12*(1-
v^2));D2:=Ew*t^3/(12*(1-v^2));D3:=Ew*t^3/(12*
> (1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/l/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=l;a2:=D2/L2^3/(1+e2);L2:=l;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
```

```
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
>
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,14..15);kg:=delcols(kg,14..15);
> kg:=delrows(kg,12..12);kg:=delcols(kg,12..12);
> kg:=delrows(kg,10..10);kg:=delcols(kg,10..10);
> kg:=delrows(kg,9..9);kg:=delcols(kg,9..9);
> kg:=delrows(kg,6..6);kg:=delcols(kg,6..6);
> kg:=delrows(kg,3..3);kg:=delcols(kg,3..3);
> kg:=delrows(kg,1..1);kg:=delcols(kg,1..1);
>
>
> kg1:=addcol(kg,7,4,1);# 13 to 7;
> kg:=delcols(kg1,7..7);
> kg3:=addrow(kg,7,4);kg:=delrows(kg3,7..7);
>
> #couple 7 and -4
> kg1:=addcol(kg,2,4,-1);
>  kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4,-1);kg:=delrows(kg3,2..2);
>
> kg1:=addcol(kg,2,4,1);# 8 to 5;
> kg:=delcols(kg1,2..2);
> kg3:=addrow(kg,2,4);kg:=delrows(kg3,2..2);
>
> # d2, d7,d8, d11
```

```
>
> f:=matrix(4,1,0);
> f[4,1]:=2*sigma*l*cos(theta);
>
> d:=matrix(4,1,0);
> d[1,1]:=delta2;d[2,1]:=delta7;d[3,1]:=delta8;d[4,1]:=delta11;
> EN:=add(multiply(2*transpose(d),kg,d),multiply(-4*transpose(d),f));
> print (EN);
> U:=multiply(2*transpose(d),kg,d);
>
# # including eo^2 terms
# #energy due to eo
> ENeo:=ky*h/2*(delta8/(2*l+h))^2+ky*h/2*((delta2-
delta11)/(2*l+h))^2+ky*l*((s2*delta2+c2*delta7-s2*
> delta8)/(2*l+h))^2-ky*l*((-s1*delta2+c1*delta7+s1*delta8)/(2*l+h))^2;
> ky:=1/2*Ew*t/(1-v^2);
> X:=EN[1,1]+2*ENeo;
>
> d2:=diff(X,delta2);
> d7:=diff(X,delta7);
> d8:=diff(X,delta8);
> d11:=diff(X,delta11);
> print (d11);
> eqnset:={d2, d7, d8, d11};
> varset:={delta2, delta7, delta8, delta11};
> solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> print (delta2);
>
> strainr:=delta11/(h+l*sin(theta));
> Et:=sigma/strainr;
>
kappa:=5/6;theta:=15/180*3.14159;v:=0.2;Ew:=1000;h:=46.078;t:=3;l:=26.9;
> c:=cos(theta);s:=sin(theta);c:=cos (theta);s:=sin(theta);
> evalf(Et);
>
Et1:=Ew*t^3*(h+l*sin(theta))/l^4/c^3/(1+(2.4+2.4*v+(tan(theta)^2+2*h/l/c
^2)*(2*l+h)/(2*l+h-v*l))*(t/l)^
> 2-v^2*((tan(theta)^2+2*h/l/c^2)*(2*l+h)/(2*l+h-
v*l))*(t/l)^2);evalf(Et1);
>
>
```

25. Maple macro- Grt, plate with shear and eo strain (psgxeo.ms)

```
> with(linalg):
> #Gibson and Ashby shear modulus derivation using simple beam element,
final version,31 July
> # 1/2 hexagonal model;
>
> keo:=v*Ew*t/(2*l+h)/(v^2-1);
>
>
> k:=matrix(15,15,0);
> ke1:=matrix(6,6,0);
> ke1[1,1]:=K+keo;ke1[1,4]:=-K-keo;
> ke1[2,2]:=12*a1;ke1[2,3]:=6*a1*L1;ke1[2,5]:=-12*a1;ke1[2,6]:=6*a1*L1;
> ke1[3,2]:=6*a1*L1;ke1[3,3]:=(4+e1)*a1*L1^2;ke1[3,5]:=-
6*a1*L1;ke1[3,6]:=(2-e1)*a1*L1^2;
> ke1[4,1]:=-K-keo;ke1[4,4]:=K+keo;
> ke1[5,2]:=-12*a1;ke1[5,3]:=-6*a1*L1;ke1[5,5]:=12*a1;ke1[5,6]:=-
6*a1*L1;
> ke1[6,2]:=6*a1*L1;ke1[6,3]:=(2-e1)*a1*L1^2;ke1[6,5]:=-
6*a1*L1;ke1[6,6]:=(4+e1)*a1*L1^2;
> print(ke1);
> T1:=matrix(6,6,0);
> T1[1,1]:=c1;
> T1[1,2]:=s1;
> T1[3,3]:=1;T1[4,4]:=c1;T1[4,5]:=s1;T1[5,4]:=-
s1;T1[5,5]:=c1;T1[6,6]:=1;
> T1[2,1]:=-s1;T1[2,2]:=c1;
> print(T1);
>
> print(ke1);
>
> print(k1);
> ke2:=matrix(6,6,0);
> ke2[1,1]:=K+keo;ke2[1,4]:=-K-keo;
> ke2[2,2]:=12*a2;ke2[2,3]:=6*a2*L2;ke2[2,5]:=-12*a2;ke2[2,6]:=6*a2*L2;
> ke2[3,2]:=6*a2*L2;ke2[3,3]:=(4+e2)*a2*L2^2;ke2[3,5]:=-
6*a2*L2;ke2[3,6]:=(2-e2)*a2*L2^2;
> ke2[4,1]:=-K-keo;ke2[4,4]:=K+keo;
>
> ke2[5,2]:=-12*a2;ke2[5,3]:=-6*a2*L2;ke2[5,5]:=12*a2;ke2[5,6]:=-
6*a2*L2;
>
> ke2[6,2]:=6*a2*L2;ke2[6,3]:=(2-e2)*a2*L2^2;ke2[6,5]:=-
6*a2*L2;ke2[6,6]:=(4+e2)*a2*L2^2;
> T2:=matrix(6,6,0);
> T2[1,1]:=c2;
> T2[1,2]:=s2;
> T2[3,3]:=1;T2[4,4]:=c2;T2[4,5]:=s2;T2[5,4]:=-
s2;T2[5,5]:=c2;T2[6,6]:=1;
> T2[2,1]:=-s2;T2[2,2]:=c2;
> ke3:=matrix(6,6,0);
> ke3[1,1]:=K2+keo;ke3[1,4]:=-K2-keo;
> ke3[2,2]:=12*a3;ke3[2,3]:=6*a3*L3;ke3[2,5]:=-12*a3;ke3[2,6]:=6*a3*L3;
>
> ke3[3,2]:=6*a3*L3;ke3[3,3]:=(4+e3)*a3*L3^2;ke3[3,5]:=-
6*a3*L3;ke3[3,6]:=(2-e3)*a3*L3^2;
> ke3[4,1]:=-K2-keo;ke3[4,4]:=K2+keo;
> ke3[5,2]:=-12*a3;ke3[5,3]:=-6*a3*L3;ke3[5,5]:=12*a3;ke3[5,6]:=-
6*a3*L3;
```

```
>
> ke3[6,2]:=6*a3*L3;ke3[6,3]:=(2-e3)*a3*L3^2;ke3[6,5]:=-
6*a3*L3;ke3[6,6]:=(4+e3)*a3*L3^2;
> T3:=matrix(6,6,0);
> T3[1,1]:=c3;
> T3[1,2]:=s3;
> T3[3,3]:=1;T3[4,4]:=c3;T3[4,5]:=s3;T3[5,4]:=-
s3;T3[5,5]:=c3;T3[6,6]:=1;
> T3[2,1]:=-s3;T3[2,2]:=c3;
>
# define a,L, I=bt^3/12, J, G, i,e1,e2,e3,K2, K,kappa
> G:=Ew/2/(1+v);Iw:=t^3/12;A:=t;D1:=Ew*t^3/(12*(1-
v^2));D2:=Ew*t^3/(12*(1-v^2));D3:=Ew*t^3/(12*
> (1-v^2));
>
e1:=12*D1/(kappa*A*G*L1^2);e2:=12*D2/(kappa*A*G*L2^2);e3:=12*D3/(kappa*A
*G*L3^2);
> K2:=Ew*A/(h/2)/(1-v^2);K:=Ew*A/1/(1-v^2);
> a1:=D1/L1^3/(1+e1);L1:=1;a2:=D2/L2^3/(1+e2);L2:=1;c2:=-
c;s2:=s;c1:=c;s1:=s;
>
> s3:=1;c3:=0;a3:=D3/L3^3/(1+e3);L3:=h/2;
> print(ke2);
> print(T2);
> print(ke3);
> print(T3);
> print(ke1);
> print(T1);
> k1:=multiply(transpose(T1),ke1,T1);
> k2:=multiply(transpose(T2),ke2,T2);
> k3:=multiply(transpose(T3),ke3,T3);
> #assembly k1
> i:=0;j:=0;n1:=7;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k1[(i+3),(j+3)];
>
> k[(n1+i-1),(n2+j-1)]:=k1[i,(j+3)];
> k[(n2+i-1),(n1+j-1)]:=k1[(i+3),j];
>
> k[(n1+i-1),(n1+j-1)]:=k1[i,j];
> od;
> od;
>
> print (k);
>
> #assembly k2
> i:=0;j:=0;n1:=4;n2:=1;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k2[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k2[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k2[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k2[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
> #assembly k3
> i:=0;j:=0;n1:=1;n2:=10;for i from 1 to 3 do
```

```
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
>
> #assembly k4
> i:=0;j:=0;n1:=13;n2:=7;for i from 1 to 3 do
> for j from 1 to 3 do
> #check ok
> k[(n2+i-1),(n2+j-1)]:=k3[(i+3),(j+3)]+k[(n2+i-1),(n2+j-1)];
>
> k[(n1+i-1),(n2+j-1)]:=k3[i,(j+3)]+k[(n1+i-1),(n2+j-1)];
> k[(n2+i-1),(n1+j-1)]:=k3[(i+3),j]+k[(n2+i-1),(n1+j-1)];
>
> k[(n1+i-1),(n1+j-1)]:=k3[i,j]+k[(n1+i-1),(n1+j-1)];
> od;
> od;
> print(k);
>
> #simplify
> i:=1;j:=1;for i from 1 to 12 do
> for j from 1 to 12 do
> ktemp:=k[i,j];
> simplify(ktemp);
> k[i,j]:=ktemp;
> od;
> od;
> print(k);
>
> s:=sin(theta); c:=cos(theta);
> print (ke2);
> print(ke1);
> print (a1);
> print(a2);
> print(a3);
# # Gk for next shear calculation
# #boundary conditions;
>
> kg:=delrows(k,13..14);kg:=delcols(kg,13..14);
> kg:=delrows(kg,11..11);kg:=delcols(kg,11..11);
>
> kg1:=addcol(kg,6,3,1);# 6 to 3;
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> #couple 8 and 5
> kg1:=addcol(kg,5,7,1);
>  kg:=delcols(kg1,5..5);
> kg3:=addrow(kg,5,7);kg:=delrows(kg3,5..5);
>
> #couple 7 and 4
> kg1:=addcol(kg,4,5,1);kg:=delcols(kg1,4..4);
> kg3:=addrow(kg,4,5);kg:=delrows(kg3,4..4);
>
> kg1:=addcol(kg,6,3,1);# 9 to 3;
```

```
> kg:=delcols(kg1,6..6);
> kg3:=addrow(kg,6,3);kg:=delrows(kg3,6..6);
>
> kg1:=addcol(kg,8,7,1);# 15 to 12;
> kg:=delcols(kg1,8..8);
> kg3:=addrow(kg,8,7);kg:=delrows(kg3,8..8);
>
> f:=matrix(7,1,0);
> # d1, d2,d3, d7, d8, d10, d12
> f[6,1]:=sigma*2*l*cos(theta);
> print(f);
> print(kg);
>
> #linsolve(kg,f);
> solutions:='';
> ur:=solutions[6,1];
> shearstrainr:=(ur)/(h+l*sin(theta));
> shearstressr:=sigma;
> #Grt:=shearstressr/shearstrainr;simplify(Grt);
>
> d:=matrix(7,1,0);
>
d[1,1]:=delta1;d[2,1]:=delta2;d[3,1]:=delta3;d[4,1]:=delta7;d[5,1]:=delt
a8;d[6,1]:=delta10;d[7,1]:=
> delta12;
> EN:=add(multiply(2*transpose(d),kg,d),multiply(-4*transpose(d),f));
> print (EN);
>
> ENeo:=ky*l*((delta1-delta7)/(2*l+h))^2;
> ky:=1/2*Ew*t/(1-v^2);
>
> X:=EN[1,1]+2*ENeo;
> d2:=diff(X,delta2);d1:=diff(X,delta1);
> d3:=diff(X,delta3);d7:=diff(X,delta7);
> d8:=diff(X,delta8);d10:=diff(X,delta10);
> d12:=diff(X,delta12);
> print (d12);
> eqnset:={d1, d2, d3, d7, d8, d10, d12};
> varset:={delta1 ,delta2, delta3, delta7, delta8, delta10, delta12};
> solutionSet:=solve(eqnset, varset);
> assign(solutionSet);
> print (delta12);
> ur:=delta10;
> shearstrainr:=(ur)/(h+l*sin(theta));
> shearstressr:=sigma;
> print (shearstrainr);
> Grt:=shearstressr/shearstrainr;
>
theta:=15/180*3.14159;v:=0.2;Ew:=1000;h:=46.078;t:=3;l:=26.9;kappa:=5/6;
> c:=cos(theta);s:=sin(theta);c:=cos (theta);s:=sin(theta);
>
> evalf(Grt);
>
Grt1:=Ew*t^3*(h+l*s)/h^2/(l*c)/(l+2*h)/(1+(t/h)^2/(l+2*h)*((h+2*l)*(l+h*
s)^2/(2*l+h-v*l)/l/c^2+2.4*h*(2*l
> +h)*(1+v)/l)-v^2*(1+(t/h)^2*(h+2*l)*(l+h*s)^2/(l+2*h)/(2*l+h-
v*l)/l/c^2));evalf(Grt1);
> print(Grt);
>
>
```

26. Cyclic constraints

(i) General Constraints

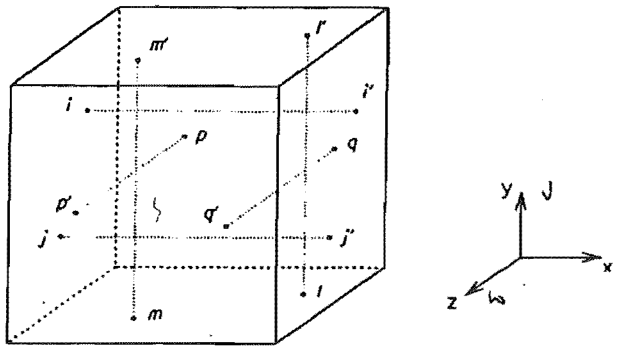


Fig. A.1: Orthogonal repeated unit

The general repeated unit shown in Fig. A.1 has x, y, and z as orthotropic axes with points i', j', l', m', p', and q' on positive faces directly opposite points i, j, l, m, p, and q on negative faces. If u, v, and w refer to displacements in the x, y, and z directions respectively, then the cyclic constraints for deformation of the unit are:

x-faces:

$$u_{i'} = u_i + \delta_x \qquad v_{i'} = v_i + \alpha \qquad w_{i'} = w_i + \beta$$
$$u_{j'} = u_j + \delta_x \qquad v_{j'} = v_j + \alpha \qquad w_{j'} = w_j + \beta$$

y-faces:

$$u_{l'} = u_l + \gamma \qquad v_{l'} = v_l + \delta_y \qquad w_{l'} = w_l + \varepsilon$$
$$u_{m'} = u_m + \gamma \qquad v_{m'} = v_m + \delta_y \qquad w_{m'} = w_m + \varepsilon \qquad \text{Eqns A.1}$$

z-faces:

$$u_{p'} = u_p + \eta \qquad v_{p'} = v_p + \xi \qquad w_{p'} = w_p + \delta_z$$
$$u_{q'} = u_q + \eta \qquad v_{q'} = v_p + \xi \qquad w_{q'} = w_q + \delta_z$$

where δ_x, α, β, γ, δ_y, ε, η, ξ, and δ_z are constants which determine the global state of strain of the structure as a whole.

In addition to these displacement constraints, rotations about each point on positive faces must be equal to rotations about opposite points on negative faces. For example, rot(i) = rot(i'). This condition holds in all of the specific cases to follow.

(ii) Specific Constraints For Normal Moduli (E_x, E_y, E_z)

Normal strains are ensured from the previous general constraints when α, β, γ, ε, η, and ξ are equated to zero, which leaves δ_x, δ_y, and δ_z as free parameters. In addition, rigid body translations are removed when $u_i = v_l = w_p = 0$. Eqns A.1 simplify to:

$$\text{x-faces:}\quad \begin{array}{lll} u_{i'} = \delta_x & v_{i'} = v_i & w_{i'} = w_i \\ u_{j'} = u_j + u_{i'} & v_{j'} = v_j & w_{j'} = w_j \end{array}$$

$$\text{y-faces:}\quad \begin{array}{lll} u_{l'} = u_l & v_{l'} = \delta_y & w_{l'} = w_l \\ u_{m'} = u_m & v_{m'} = v_m + v_{l'} & w_{m'} = w_m \end{array}$$

$$\text{z-faces:}\quad \begin{array}{lll} u_{p'} = u_p & v_{p'} = v_p & w_{p'} = \delta_z \\ u_{q'} = u_q & v_{q'} = v_p & w_{q'} = w_q + w_{p'} \end{array}$$

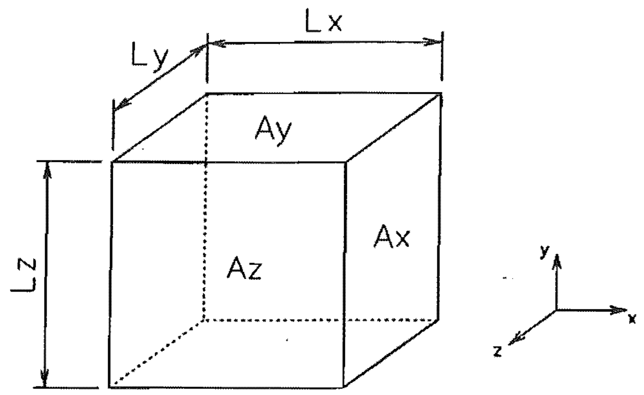


Fig. A.2: Dimensions of the general unit

If L_x, L_y, L_z, A_x, A_y, and A_z are defined as shown in Fig. A.2. then normal moduli can be evaluated simply by applying the following independent loads.

(1) Apply unit tensile load in x-direction to point i' to give

$$E_x = \frac{L_x}{\delta_x A_x}$$

$$\nu_{xy} = -\frac{\delta_y L_x}{\delta_x L_y}$$

$$\nu_{xz} = -\frac{\delta_z L_x}{\delta_x L_z}$$

(2) Apply unit tensile load in y-direction to point l' to give

$$E_y = \frac{L_y}{\delta_y A_y}$$

$$\nu_{yx} = -\frac{\delta_x L_y}{\delta_y L_x}$$

$$\nu_{yz} = -\frac{\delta_z L_y}{\delta_y L_z}$$

(3) Apply unit tensile load in z-direction to point p' to give

$$E_z = \frac{L_z}{\delta_z A_z}$$

$$\nu_{zx} = -\frac{\delta_x L_z}{\delta_z L_x}$$

$$\nu_{zy} = -\frac{\delta_y L_z}{\delta_z L_y}$$

(iii) Specific Constraints For The Shear Modulus G_{yx}

Shear strain in the x-y plane is obtained by setting $\delta_x = \delta_y = \delta_z = \beta = \varepsilon = \eta = \xi = 0$ while rigid body translations and rotations are removed by setting $u_l = v_i = w_p = 0$. If $\alpha = 0$ then γ is the only free constant and thus Eqns A.1 simplify to

x-faces:	$u_{i'} = u_i$ $u_{j'} = u_j$	$v_{i'} = v_i = 0$ $v_{j'} = v_j$	$w_{i'} = w_i$ $w_{j'} = w_j$
y-faces:	$u_{l'} = \gamma$ $u_{m'} = u_m + u_{l'}$	$v_{l'} = v_l$ $v_{m'} = v_m$	$w_{l'} = w_l$ $w_{m'} = w_m$
z-faces:	$u_{p'} = u_p$ $u_{q'} = u_q$	$v_{p'} = v_p$ $v_{q'} = v_p$	$w_{p'} = w_p = 0$ $w_{q'} = w_q$

Applying a unit load in the positive x-direction to l', the shear modulus G_{yx} is given by

$$G_{yx} = \frac{L_y}{\gamma A_y}$$

(iv) Specific Constraints For The Shear Modulus G_{yz}

Shear strain in the y-z plane is obtained by setting $\delta_x = \delta_y = \delta_z = \alpha = \beta = \gamma = \eta = 0$ while rigid body translations and rotations are removed by setting $u_p = v_i = w_l = 0$. If $\xi = 0$ then ε is the only free constant and thus Eqns A.1 simplify to

$$\text{x-faces:} \qquad \begin{array}{lll} u_{i'} = u_i & v_{i'} = v_i = 0 & w_{i'} = w_i \\ u_{j'} = u_j & v_{j'} = v_j & w_{j'} = w_j \end{array}$$

$$\text{y-faces:} \qquad \begin{array}{lll} u_{l'} = u_l & v_{l'} = v_l & w_{l'} = \varepsilon \\ u_{m'} = u_m & v_{m'} = v_m & w_{m'} = w_m + w_{l'} \end{array}$$

$$\text{z-faces:} \qquad \begin{array}{lll} u_{p'} = u_p = 0 & v_{p'} = v_p & w_{p'} = w_p \\ u_{q'} = u_q & v_{q'} = v_p & w_{q'} = w_q \end{array}$$

Applying a unit load in the positive z-direction to l', the shear modulus G_{yz} is given by

$$G_{yz} = \frac{L_y}{\varepsilon A_y}$$

(v) Applying Constraints in ANSYS 5.0

The constraint relationships set out in the previous subsections of this Appendix apply to the modelling of any repeated unit, irrespective of the analysis method. To apply these constraints to an ANSYS model there must first be an understanding of the limitations involved in the ANSYS constraint methods, particularly with respect to the concepts of 'Fixed degrees of freedom', 'Coupled sets', and 'Constraint equations'.

A *fixed DOF* is a translation (UX, UY, UZ) or rotation (ROTX, ROTY, ROTZ) of a particular node which is set to zero.

A *coupled set* contains the DOFs of two or more nodes, causing them to take on the same (but unknown) value. The same DOF cannot appear in more than one coupled set, and no set can contain a fixed DOF.

A *constraint equation* is a linear relationship between any two or more DOFs. It is more general than a coupled set, and takes the form:

$$\text{Constant} = \sum_{i=1}^{N} (\text{Coeff.}(i) \times \text{DOF}(i))$$

where DOF(i) is the degree of freedom of term (i), and N is the number of terms in the equation. No term may contain a coupled DOF, and there must be at least one unique DOF in the equation - that is, one DOF which does not appear in any other constraint

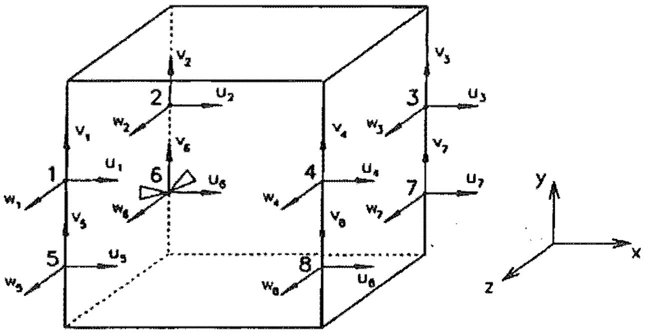


Fig. A.3. Orthogonal repeated unit with edge nodes

equation.

The above ANSYS principles pose limitations on the ways in which the constraints can be applied to a model. As an example, consider the repeated unit model shown in Fig. A.3. Nodes 1, 2, 3, and 4 lie on the same x-z plane as do nodes 5, 6, 7, and 8, being on the edges of the model. If the x and z DOFs of node 6 are then fixed (setting $u_6 = w_6 = 0$) as shown, then from the general constraint relationships from section (ii) of this appendix, for normal strain, we have:

x-faces:

$u_4 = u_1 + u_7$	...(1)	$v_4 = v_1$	...(5)	$w_4 = w_1$	...(9)
$u_3 = u_2 + u_7$	...(2)	$v_3 = v_2$	...(6)	$w_3 = w_2$	...(10)
$u_8 = u_5 + u_7$	...(3)	$v_8 = v_5$	...(7)	$w_8 = w_5$	...(11)
$u_7 = \delta_x$	...(4)	$v_7 = v_6$	...(8)	$w_7 = w_6$	...(12)

z-faces:

$u_1 = u_2$	...(13)	$v_1 = v_2$	...(17)	$w_1 = w_2 + w_5$	...(21)
$u_4 = u_3$	...(14)	$v_4 = v_3$	...(18)	$w_4 = w_3 + w_5$	...(22)
$u_5 = u_6$	...(15)	$v_5 = v_6$	...(19)	$w_5 = \delta_z$	...(23)
$u_8 = u_7$	...(16)	$v_8 = v_7$	...(20)	$w_8 = w_7 + w_5$	...(24)

These however cannot be applied in ANSYS without modification. Consider the limitations in turn with associated examples:

1. *No coupled set may contain a fixed DOF.* Hence from (15) and $u_6 = 0$ being previously set, another fixed DOF must be created, giving

$$u_5 = 0$$

2. *No DOF may appear in more than one coupled set.* Hence (5), (6), (17), and (18) - which relate the same four DOFs - must be merged, giving

$$v_1 = v_2 = v_3 = v_4$$

3. *No constraint equation may contain a coupled DOF.* Equations (1), (2), and (3) clearly must be written as constraint equations. Equations (13), (14), and (15) contain DOFs which also appear in these constraint equations and hence cannot be written as coupled sets.

4. *Each constraint equation must contain at least one unique DOF.* Equations (1), (2), (13), and (14) violate this rule because each DOF that they relate (u_1, u_2, u_3, u_4, and u_7) appears in more than one of these equations. They must be rewritten in order that the first term in each is unique. Note that three equations result since the fourth is redundant.

$$0 = u_4 - u_1 - u_7$$
$$0 = u_3 - u_1 - u_7$$
$$0 = u_2 - u_1$$

In summary, the final ANSYS constraints are

[1] *Fixed DOF* constraints:

$u_5 = 0$	$w_6 = 0$
$u_6 = 0$	$w_7 = 0$

[2] *Coupled set* constraints:

$$v_1 = v_2 = v_3 = v_4$$
$$v_5 = v_6 = v_7 = v_8$$

[3] *Constraint equation* constraints:

$0 = u_4 - u_1 - u_7$	$0 = w_1 - w_2 - w_5$
$0 = u_3 - u_1 - u_7$	$0 = w_4 - w_2 - w_5$
$0 = u_2 - u_1$	$0 = w_3 - w_2$
$0 = u_7 - u_8$	$0 = w_8 - w_5$